70-50397

DRF

THE MITRE CORPORATION
Bedford, Massachusetts

MTR-1202

MISSION CONTROL SYSTEMS EFFECTIVENESS ANALYSIS

10 JUNE 1966

Volume III
THE FINAL REPORT
RTCC & SCATS ANALYSES

NASA CONTRACT T-53803-G

MITRE Technical Report

MTR- 1202 3

No. Vol. Series Rev. Supp. Corr.

Subject: Part I: Real Time Computer Complex (RTCC)
Part II: Simulation Checkout and Training System (SCATS)

Author: Part I: A. Cohen, E. S. Herndon
Part II: D. I. Buckley

Dept.: D-05

Date: 10 June 1966

Contract No.: AF 19(628)-5165

Contract Sponsor: ESD

Project: 8270

Issued at:

Department Approval: W B Woodward
W. B. Woodward

MITRE Project Approval: Daniel Goldenberg
D. Goldenberg

THE MITRE CORPORATION

BEDFORD, MASSACHUSETTS

Page 1 of 78 Pages

ABSTRACT

PART I: RTCC

The evaluation of the Real Time Computer Complex presented in this study concentrates on three general objectives:

1. Capacity and Capability of the System to Meet Requirements
2. Effectiveness with which the System Meets these Requirements
3. Capability of the System to Respond to Changing Requirements

Emphasis is placed on the organization of the system, particularly of the Executive program, and the use of storage, and how these relate to the above evaluation objectives.

Part II: SCATS

This part sets forth the results of the initial study of the Simulation Checkout and Training System, defines certain salient technical questions that remained unanswered at the conclusion of the study and recommends analyses that should be conducted in the SCATS and Apollo SCATS to provide answers to these questions.

A. Cohen

A. Cohen

E. S. Herndon

E. S. Herndon

D. I. Buckley

D. I. Buckley

TABLE OF CONTENTS

PART I

TABLE OF CONTENTS (CONTINUED)

PART I

REAL TIME COMPUTER COMPLEX ANALYSIS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

BACKGROUND OF STUDY

This part of the MITRE study has been directed at the present IBM 7094 System, notwithstanding the fact that this system is being replaced by an IBM 360-75 System. The first 360-75 has been installed already and the whole of the IBM 7094 System will be replaced by the end of 1966. At the time at which this study began, there was little information available about the 360-75 System which could form the basis for analysis. At the direction of NASA, MITRE has therefore studied and analyzed the present capability, and no time has been spent on the IBM 360-75 System.

The study began with a detailed analysis of the IBM 7094 computers and the software system. It soon became apparent that the key to understanding the software lay in the Executive System. The Executive is fundamental to the performance and operation of the system. This phase was completed by the generation of two sets of charts describing the working of the Executive - the Executive System Functional Chart, which describes the Executive from a functional point of view, and three Executive System Logic Charts, which chart the logic flow of control through the Executive.

This analysis of the Executive has been followed by analyses of selected topics, the selection being determined primarily by the time available to complete the study.

GENERAL OBJECTIVES OF THE STUDY

Three general objectives have been defined for a study of the RTCC. They are:

1. Capacity and Capability of the System to Meet Requirements
2. Effectiveness with which the System Meets these Requirements
3. Capability of the System to Respond to Changing Requirements

GENERAL OUTLINE OF STUDY

Section II of this part of the report provides a general functional description of the RTCC System; Section III describes the objectives of the study; Section IV contains the analysis; Section V summarizes the principal conclusions; and Section VI discusses recommendations for future study.

SECTION II

GENERAL FUNCTIONAL DESCRIPTION

INTRODUCTION

A general functional description of the RTCC system is provided in this section to introduce those system elements for which data is presented in the analysis section which follows. The principal emphasis in the analysis is on the organization of the system and the performance characteristics of the executive software; consequently in this description the material is presented on both hardware and program functions as they relate to the analysis objective. The mechanization of these functions is treated only to that level of detail which is required to understand what each element of the system does. It is assumed that the reader has a general working knowledge of the RTCC. While no particular orientation to hardware or software is assumed, those associated with the latter may find this report of greater interest.

RTCC HARDWARE FEATURES

This section discusses certain features of the RTCC hardware system which have an important bearing on the software. No attempt is made, however, to describe the hardware comprehensively.

Figure 3-1 shows a simplified block diagram of one of the IBM 7094 computers. Table A-3 in Appendix A contains a list of the principal components of the computer.

Multiprogramming Capability

The multiprogramming capability of the RTCC System is facilitated by two hardware features the use of which are under program control. These are described very briefly.

Address Relocate Mode

When in the Address Relocate Mode, an 8-bit Relocate Register contains a Relocation Factor which is added to all CPU-generated memory addresses, except those generated during the execution of a trap. The Relocate Register does not contain the least significant 8 bits of the address, so that relocation takes place by a multiple of 256, up to and including 255 X 256 (65,280). The computer can enter or leave the

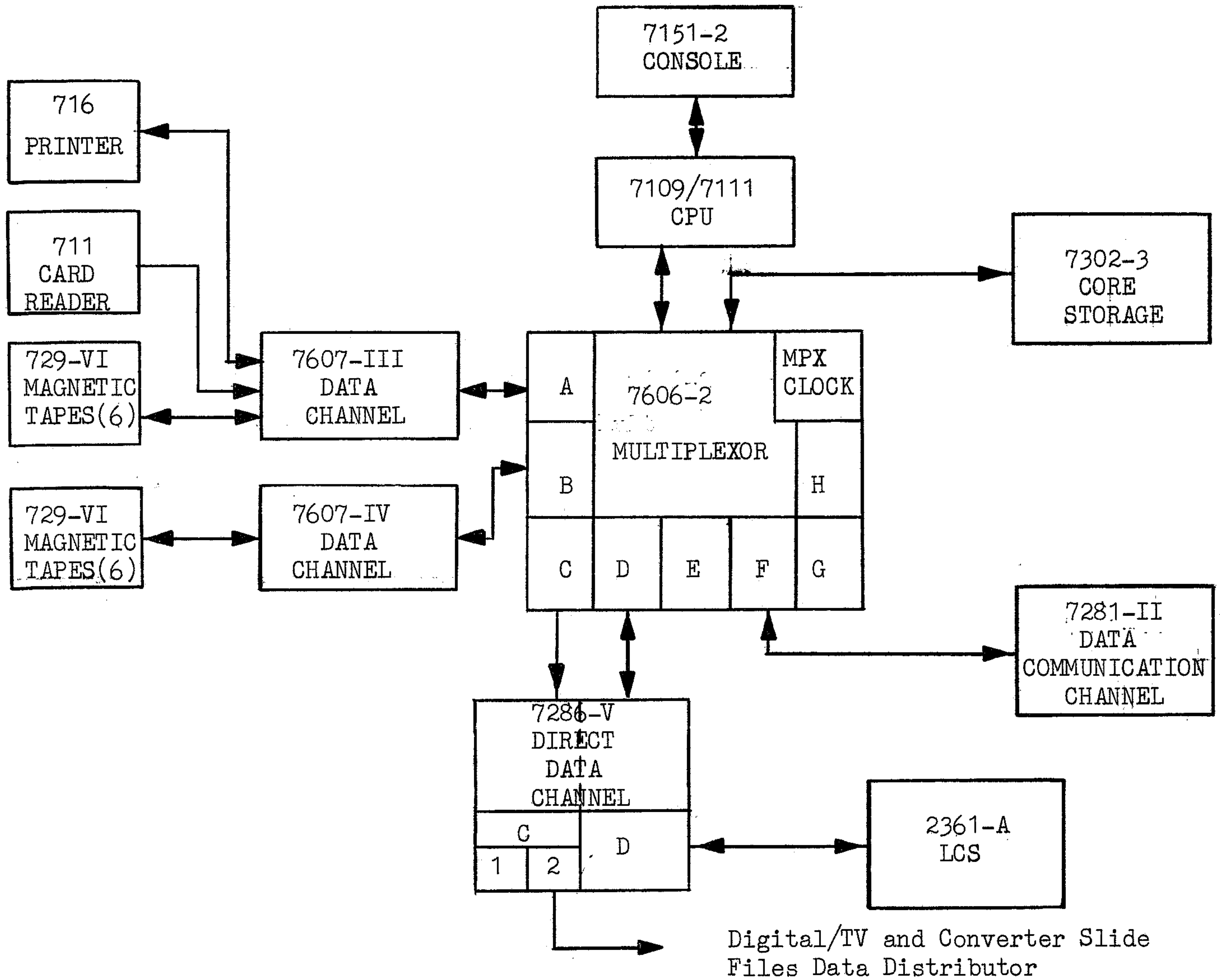


Figure 3-1 IBM 7094 - II - Block Diagram

Relocate Mode under program control, and it will also leave the Relocate Mode on a trap, or upon the execution of the Store and Trap (STR) instruction.

The use of the Relocate feature allows all programs, except the Executive Program that resides permanently in main core, to be assembled relative to location zero, but executed from any area of core that is available at the time of execution.

Address and Instruction Protection Mode

The Address and Instruction Protection Mode (AIPM) permits areas of storage to be protected. Two registers, an Upper Bounds Register (UBR) and a Lower Bounds Register (LBR), are loaded with the bounds of an area of storage, and the computer enters AIPM. Either "inside protection" or "outside protection" may be specified. Inside protection protects the area bounded between the LBR and the UBR, and outside protection protects the area outside these bounds. A "Protect Trap" will occur when any address used to access main memory falls within the protected area, unless it is an I/O generated address, or an address generated by a trap or the STR instruction. The computer can enter or leave the Address and Instruction Protection Mode under program control, and it will also leave AIPM on a trap or upon executing the STR instruction.

The use of the AIPM feature permits the Executive to restrict a program to its own area of core, and to prevent it from altering anything outside its own area. Thus the integrity of all core outside the program area is guaranteed.

IBM 2361-A Large Capacity Storage

This core-storage device has been referred to throughout this report as the "Core-File." It should not be confused with the IBM 7253 Core Storage File of 262,144 words which was formerly used.

The 2361-A Core-File provides 524,288 words of random access storage. The data transfer rate is 250,000 words/second through the IBM 7286 Direct Data Channel. Data may be transferred between main core and the Core-File in single blocks, of any size, within the size of main core available, starting at any location in the Core-File, to any area in main core.

PROGRAM STRUCTURE

Two aspects of what may loosely be called program structure are discussed in this section: the use of storage and the division of program tasks. The program may be grossly divided into two classes of programs: Executive programs and Mission programs. The division of program tasks between these two classes of programs will be described. In addition, the allocation of storage to these two classes of programs and their associated tables is discussed. Emphasis is placed on the services provided by Executive to the Mission programs and how storage is used in the system because these will be discussed extensively in the analysis section which follows later.

Use of Main Core and Core-File Storage

All programs are operated from main core; in it reside the input and output buffer areas associated with the Data Communication Channel (DCC), the most frequently used routines of the Executive program, and an Executive buffer pool which is used for temporary storage of data and outstanding program processing tasks. The remainder of main core is dynamically allocated to programs which reside on the Core-File and are called in to operate as required.

The Core-File is used as a high speed random access bulk storage device for programs and data tables. All Mission programs and some of the less frequently used Executive routines are stored on the Core-File.

Division of Program Tasks

The program system is built on a concept of dividing the tasks between an Executive, most of which resides permanently in main core, and Mission programs which are stored on the Core-File and brought into main core when they are required to operate.

The Executive performs several tasks for the system. It handles all input and output operations, provides for logging of data, allocates space in core for programs which are needed, and calls programs into operation on the basis of requests by other programs or the receipt of inputs. With respect to this last function, the Executive maintains a program priority and status table which indicates for each program where it is located and where its outstanding tasks are located in the Executive buffer pool. These tables also indicate by control bits which programs are queued (that is, have an outstanding task), which programs are in process, and which programs are suppressed from operation. These Executive control functions are discussed in greater detail in the section following, EXECUTIVE SYSTEM FUNCTIONS.

Mission programs, which are relocated and protected, rely heavily on the Executive. They communicate with one another via the Executive. They rely on the Executive for all input/output operations. Mission programs, because of the relocate and protect features, cannot make references to absolute locations nor can they request memory accesses outside their protected area. In order to obtain data, the Mission programs request the Executive to place the data in their work area. Mission programs also rely on the Executive for such miscellaneous services as clock readings and the formatting of data for BCD or teletype output.

The Mission programs contain all of the logic required to control the mission and to provide the computation associated with tracking, telemetry, etc. The control decisions are performed by programs called supervisors and their associated "functions."* The supervisors call upon processors, a second class of programs, to perform necessary calculations. Thus virtually all mission related activity is contained in the Mission programs.

The Executive system, on the other hand, has very little cognizance of mission activity. This cognizance is restricted to a routing table which identifies each incoming message type and indicates how the data should be handled, and the program priority table which indicates the relative priority and status of each of the Mission programs.

---

* "Functions" when used with quotation marks will indicate a unit of operating code which is part of a supervisor. This is done to make clear that we are talking of "functions" as used by IBM in their RTCC program literature. The word function without the quotation marks will be used in its less restrictive sense throughout this section.

EXECUTIVE SYSTEM FUNCTIONS

Figure 3-2 presents a block diagram indicating the flow of information controlled by the Executive system. A brief description of each of these Executive control functions will be presented.

1. Data Communication Channel Inputs - Upon a trap by the DCC, the Executive takes control and moves the input data to the Executive buffer pool. It then interprets the message identifier portion of the input message. Three options exist for the disposition of input messages; they may be ignored, stored on the Core-File, or routed to the appropriate mission program. In the last option, routed inputs, the Executive leaves the data in the Executive buffer pool along with appropriate control information and queues the mission program which will process the data.

> NOTE: At the end of each of the functions described in this section, the Executive enters a sequencing routine to determine which program to operate next. The routine has three entries. At the principal entry it searches the priority table for the highest priority program which is queued or in process and ready to operate. The other two entries to the sequencing routine bypass the search of the priority table; these are used when the priority of a program can be assumed. These entries either restore the last program to operate, or restrict the priority search to "functions" within a given supervisor. At the end of the sequencing routine the most current task is moved to the program working area and the program is started.

2. DCC Outputs - Mission programs which have prepared data for output on the Data Communication Channel call upon the Executive to do the actual output of data. This activity of the Executive includes moving the data to the output buffers and initiating the channel commands. If the subchannel is busy, the request is stacked until the channel traps, thereby indicating completion of the last output.

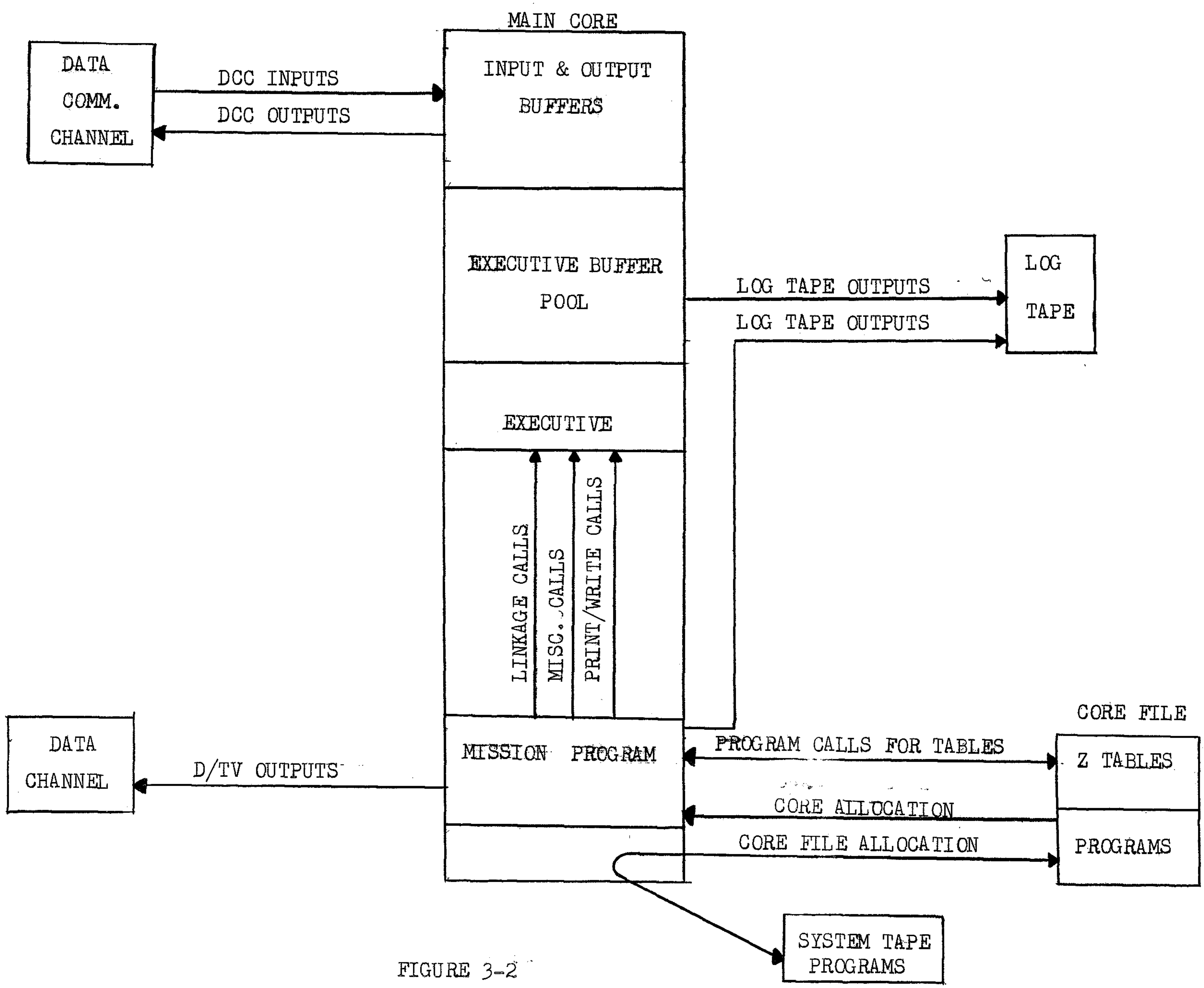


FIGURE 3-2

FLOW OF INFORMATION CONTROLLED BY EXECUTIVE

3. <u>Digital to TV Converter Outputs</u> - This activity of the Executive system is analogous to that for DCC outputs with the difference that the outputs are made from program working areas instead of the dedicated buffer areas.

4. <u>Log Tape Outputs</u> - The logging activity of the Executive is used to record on tape selected DCC inputs, DCC outputs, and Digital/TV Converter Outputs. The logging activity consists of routines to create log records, to initiate the output, and, upon completion of output, to make available the Buffer areas and/or program areas from which data was written.

5. <u>Linkage</u> - Mission programs communicate with one another indirectly through the Executive linkage routines. The Executive interprets a wide variety of linkage calls. These calls are used to queue new programs, pass control information or data to other programs via the Executive, or to indicate completion of the previous task.

6. <u>Miscellaneous Services</u> - The Executive provides several services for mission programs upon request. These include: suppression of "functions," release of "functions," moving data into a designated area of a supervisor, changing the input routing table, providing clock readings and calls for programs on the system tape to be read onto the Core-File.

7. <u>Print/Write Services</u> - Upon request of mission programs the Executive will take data from a mission program, format it for BCD or teletype and output the data on the appropriate channel.

8. <u>Input/Output of Z Tables</u> - In the course of operation, mission programs frequently require data tables, known as Z Tables, which are stored on the Core-File. These I/O services are handled by the same Executive routines which handle all other I/O operations, the only reason for distinguishing between the classes of I/O is to indicate the different types of traffic involved.

9. <u>Core Allocation</u> - The Executive receives input data and linkage calls from mission programs both of which can result in a need for a program which is not currently in main core. Core allocation encompasses those Executive activities which are required to find space in

main core, move the program into core from the Core-File and enter the program.

10. Core-File Allocation - It is possible during some missions that all of programs required cannot be accommodated in Core-File and that some must remain on the system tape. To provide for such cases, the Executive has routines to move programs to Core-File from the system tape via main core.

SECTION III

STUDY OBJECTIVES

EVALUATION OF SALIENT STUDY AREAS

This section presents a brief survey of the three salient questions which are considered important for analysis in a study of the RTCC.

1. Capacity and Capability of the System to Meet Requirements
2. Effectiveness with which the System Meets these Requirements
3. Ability of the System to Respond to Changing Requirements

Capacity and Capability of System to Meet Requirements

A principal question of interest is: "Can the job be done within the time constraints of the real-time environment?" The ability of a system to perform a job is a function of the computational load imposed by the job, the performance characteristics of the hardware and the individual system programs, and the way in which the system is organized to handle the programs and data. The question of performing the job within the time constraints will resolve into a consideration of whether the required computations are performed without excessive backlogging or delays. System loading, expressed as the percentage of time that the central processing unit is required to work, is a measure of whether jobs are being done within time constraints. If for several seconds the system is saturated, that is, if loading is at 100% capacity, jobs are being backlogged and delayed.

The system loading varies during the mission as the computing requirements change from phase to phase. An analysis of system loading would consider both the variations in computational load and the computer hardware and software performance characteristics. This type of analysis has been done in the past by IBM to estimate the capability of the RTCC to perform the necessary computations. A computer system model was used to perform this analysis in which machine configurations as well as input loading characteristics were varied to determine if the system would carry the load.

Another question of interest is concerned with the utilization of storage capacity, including the use of main core, Core-File and magnetic tapes. The amount of core storage available clearly has an important bearing on the capacity and capability of the system and it influences the usage of CPU capacity in subtle ways. The manner of use of the Core-File is also important not only as it determines the effective total storage available, but also in its influence on the use of CPU capacity.

Effectiveness With Which the System Meets These Requirements

No less important than the study of capacity and capability of the system and an essential adjunct to it, is a study of system effectiveness. The type of questions addressed in such a study are not whether the system does the job, but how well it does it. This study focuses on the organization of the system to determine if savings in computing capacity can be achieved by changing the way that programs and data are handled. This study would consider in particular the design characteristics of the Executive System and would make use of data on the loading imposed by various parts of the system.

Ability of the System to Respond to Changing Requirements

The flexibility of the system, or ability to respond to changing or additional requirements, is perhaps one of the most controversial areas of interest. It is not a question that is readily amenable to quantitative evaluation - rather it is more a subject for opinions and value judgments. However, it is of direct concern to many people outside the RTCC and as such it should be worthy of study.

SCOPE OF THE MITRE STUDY

In seeking answers to each of the salient questions for study, the analyses that were made are described as follows:

1. Capacity and Capability of the System to Meet Requirements

The study has been restricted on this question to an analysis of the structure of program and data storage on the Core-File and a discussion of how the effective storage capability can be increased.

2. Effectiveness with which the System Meets these Requirements

Results from system model simulations as well as the statistics taken during mission simulations have indicated that a large portion of the system capacity is taken up by the Executive System, roughly half of the used computing capacity. This observation led to an examination of how the system is organized with particular emphasis being placed on the functions performed by the Executive System. The effect of system organization on system loading has not been treated in detail in the past, and so it was felt that by concentrating on the system organization some new light could be shed on the impact of software design on system computing capacity.

3. Ability of the System to Respond to Changing Requirements

This part of the MITRE study has been restricted to a brief survey of the important characteristics of software that allow flexibility and of the principal aspects of the hardware and software systems that influence the ability to respond to changing requirements.

SECTION IV

ANALYSIS

INTRODUCTION

This section reports the analysis of three topics which have been covered by the MITRE study of the RTCC System. They are:

1. Core-File Utilization
2. System Loading
3. System Flexibility

CORE-FILE UTILIZATION

This section presents a discussion on the structure of Core-File storage. In this context "structure" refers to the way in which the contents of the Core-File can be related to the phases of the mission.

Current practice is to store all programs and all data tables (Z Tables) on the Core-File for all phases of the mission. The capability exists to store programs on magnetic tape, and to bring them in to the Core-File as required (queued) by the program. When a program, that is not on the Core-File, is queued, space is allocated on the Core-File and the program is read in from magnetic tape and written into the allocated space. Information on the requirements for programs by phase of the mission is readily available. At present the software capability does not exist to allocate Z Tables from tape as needed and information on the requirements for Z Tables by phase of the mission is not readily available.

The approach to the analysis of mission program storage was to classify them according to mission phase in which they are used, and to attach a "Usage Code" to each mission program. This Usage Code will be described under "Usage of Mission Programs", and it has a bearing on a discussion of the use of magnetic tape as an auxiliary store to back up the Core-File. This will be shown, in the paragraphs under "Discussion of Mission Program Storage Requirements", later in this section.

The approach to analysis of data table storage was to classify and tabulate data tables by size of the table (number of words). The classification of data tables is not of primary importance to a discussion of the structure of the Core-File storage, but it has a bearing on certain aspects of the analysis of Program Operating Statistics in the section on System Loading.

Sources of Data

The analysis of Core-File Utilization was based on data for the GTA-9 Gemini-Agena Rendezvous Mission. The following sources of data were used:

1. "Ninth Gemini Mission Development Plan", dated 3 May 1966. This document provided a list of programs included in the GTA-9 Mission Operational Program.

2. "Gemini-Apollo Executive System Initialization. Copies of On-Line Messages and Other Information", dated 6 May 1966. This on-line printout, made at the time of Executive Initialization, provided information on the sizes of programs on the Core-File, and a list of Z Tables on the Core-File and their dimensions, for GTA-9.

3. Discussions with James C. Stokes, Assistant Chief, Real-Time Program Development Branch, Mission Planning and Analysis Division. Information was obtained on the usage of programs and the phases in which they are used.

Analysis of Core-File Utilization by Mission Phase

Phases of the Mission

A Gemini rendezvous mission, such as GTA-9, is divided into the following phases:

Prelaunch 1 - Agena prelaunch (P1)
Launch 1 - Agena launch (L1)
Prelaunch 2 - Gemini prelaunch (P2)(Includes Agena Orbit)
Launch 2 - Gemini launch (L2)
Orbit - Gemini and Agena orbit (0)
Reentry - Gemini reentry (R)

Programs Required in Mission Phases

The accompanying Table 3-1 summarizes the requirements for storage on the Core-File in each phase of the mission. Table A-1 in Appendix A lists these requirements in more detail.

It may be observed from Table A-1, that the Prelaunch 2 and Orbit phases overlap to a very great extent. Many of the Orbit phase programs are required in Prelaunch 2 because the Agena is in Orbit. This processing is suspended during Launch 2. During Launch 2, however, there is a requirement to have Re-entry phase programs on the Core-File, in case of an abort. Therefore, Table 3-1 includes a column which combines the requirements for Launch 2 and Re-entry.

Usage of Mission Programs

Table A-1 contains the following code (Usage Code) to classify the usage of mission programs:

- F - Frequently used
- I - Infrequently used
- T - Used at particular times during a mission or during an orbit, but not at regular frequent intervals, or upon a manual input
- C - Contingency use during abort

T is always used in combination with F or I. The implication of T is that the program would probably not be required at short notice. Either the time at which these programs are needed would be defined well in advance, or the need is a result of some decision by a controller, and in such cases there is probably no great urgency (not required within a fraction of a second).

The combination FT means that the program is used only at certain times, which may occur infrequently, but when the occasion arises the program is heavily used. The combination IT means that the program is used only at certain times, and, at these times, it is used infrequently.

TABLE 3-1 SUMMARY CORE-FILE STORAGE REQUIREMENTS FOR GTA-9

Mission Phase	P1	L1	P2	L2	O	R	L2+R
Program Storage Required	102,568	117,618	294,626	137,638	307,184	134,798	175,140
Z Table Storage Required	124,501	124,501	124,501	124,501	124,501	124,501	124,501
TOTAL	227,069	242,119	419,127	262,139	431,685	259,299	299,641

No frequency code is specified for Executive routines. That part of the Executive which resides in core contains the most frequently used routines, while some of the routines on the Core-File are comparatively infrequently used.

Discussion of Mission Program Storage Requirements

Table 3-1 indicates that the largest requirements for Core-File storage occur during the Prelaunch 2 and Orbit phases. The total storage requirement for programs and tables for all phases of the mission is 497,162 while the maximum requirement occurs during the orbit phase, when it is 431,685. It is clear, therefore, that no great saving of Core-File storage can be achieved simply by allocating programs from tape to the Core-File as required by phase. This procedure would reduce the maximum Core-File storage requirement, or otherwise increase the amount of available Core-File storage, by about 63,793 (note: this includes the need for two additional programs on the Core-File - the Core-File Allocation Supervisor, and the Tape-to-Core-File Transmission Processor).

In order to increase still further the available Core-File storage, it would be necessary to remove selected routines from the Core-File, and to bring each program in from tape when it is queued. The kind of programs that would be prime candidates for removal from the Core-File, would be those that are used only at particular times - indicated by a T under Usage Code in Table A-1. The assumption is that the need for one of these programs can be established early enough so that the program can be transferred from tape to the Core-File in time to be used. Alternatively a controller makes a decision that the time is right to carry out this processing; if there is no urgency, some delay can be tolerated while the program is brought in. Of course, it would be necessary to consider each program on its own merits to decide whether it could be satisfactorily used from tape.

Core-File Allocation has been used in the past and the technique will work. However, it appears that the procedure was not entirely satisfactory. The problems with the procedure were mostly associated with the time it takes to change over from one phase to another, or the time to bring in a single routine from tape to Core-File. In part, this was no doubt due to interruptions for higher priority processing. If there were a real need to use this feature of the system again, the recently improved system could be made to work satisfactorily. However, it is worth pointing out that it is not just a programming problem. Strictly from a programming point of view, it is possible to change over completely from one phase to another within 15 to 20 seconds. However, this would imply that mission controllers would accept some degradation of service, even to nearly complete suspension of processing during the transfer. Some changes to Executive would probably be necessary, but no analysis has been made of this in this study.

Analysis of Table Storage

Data Tables (Z Tables) are stored permanently on the Core-File, and as was stated earlier, no capability exists at present to allocate tables to the Core-File by mission phase or when needed. Data, that would indicate the phases in which a table is used, is not readily available, and no information has been found to indicate that allocation of tables to Core-File by phase or by need would result in significant saving of Core-File space. The analysis of table storage has therefore been restricted to a tabulation of tables by length of table (number of words). Table A-2 in Appendix A contains a full listing of number of tables against length of table. This data is summarized in Table 3-2 in groups of table size. Thus for example, the first size group includes all tables between 1 and 25 words in length. Table 3-2 lists "Number of Tables" in each group and the "Size of Group" (total number of words of storage required). Table 3-2 also lists "% Number of Tables", "% Size of Group", and cumulative percentages, of the whole set of Z Tables.

Table 3-2 indicates clearly that the majority of Tables are quite short. While in itself this fact is not necessarily very significant, the data will be used again in the section on System Loading.

Summary of Core-File Utilization Analysis

This analysis has indicated that at the time of maximum requirement for Core-File storage, which occurs during the Orbit Phase of the mission, approximately 87% of the total programs and table storage is needed. This is based on the assumption that all Z Tables are required during all phases of the mission. Thus if programs were allocated to the Core-File by phase of the mission, 63,793 words of Core-File storage would be saved. Further saving of space on the Core-File could be made by removing selected programs, and allocating them from tape to the Core-File when needed.

TABLE 3-2 Z TABLES - FREQUENCY SUMMARY IN GROUPS

Size Group	No. of Tables in Group	Size of Group	% No. of Tables	% Size of Group	Cum. % No. of Tables	Cum. % Size of Group
1-25	55	716	19.4	0.6	19.4	0.6
26-50	52	1,973	18.4	1.6	37.8	2.2
51-100	44	3,343	15.6	2.6	53.4	4.8
101-200	33	5,323	11.6	4.3	65.0	9.1
201-400	38	10,829	13.5	8.7	78.5	17.8
401-800	32	19,200	11.3	15.4	89.8	33.2
801-1600	13	14,885	4.6	12.0	94.4	45.2
1601-3200	9	19,229	3.1	15.4	97.5	60.6
3201 up	7	49,123	2.5	39.4	100.0	100.0
TOTALS	283	124,621				

It has been found that the majority of tables on the Core-File are quite short. This aspect of Core-File Utilization will be examined in the section on System Loading.

SYSTEM LOADING

The discussion of system loading will attempt to dissect the system load and identify how much of the load is due to each functional element of the system. The purpose of this analysis is not to identify the cause of changes in system load as the mission progresses from phase-to-phase, but rather to look at a profile of system utilization and indicate how much of system capacity is used for mission programs and how much is used by each of the functional elements of the executive.

Sources of Data

Data for the system loading analysis was taken from an APOLLO 201 simulation mission. The data was taken on 31 January 1966 using the GSSC to provide inputs to the system. Mission control personnel participated in the conduct of the test; therefore, realistic display request activity is included in the system load.

Although it would have been more desirable to use data from one of the GEMINI missions, this was not possible, because executive statistics have not been recorded on GEMINI missions since GT-3 and GT-4. In the interim period, several major changes to the system have been made thereby making the performance data taken on those early missions less applicable to the current system. These changes include doubling the size of main core and the deletion of an Executive processor called XXRUTE, which was replaced by a more efficient system of handling store mode inputs. Attempts to obtain data on the GT-9 Launch Abort simulations were not successful.

The data taken during the APOLLO 201 simulation includes two of the three groups of statistics which can be taken, by the system, namely, Central Processor Utilization and the Executive Statistics. The third group, Processor Statistics, were not included in the run.

The Central Processor Unit (CPU) utilization statistics present three measurements of system loading: the percentage of time that the system was being used, the percentage of time that the system was waiting for I/O and the percentage of time that the system was idle.

The Executive statistics present, for each of approximately eighteen Executive activities, the average time per execution, the frequency of execution and the percent of central processor time used by the activity. It should be noted that these Executive activities are not identical with the Executive functions indicated in the Executive functional description, but, the loads measured for the individual activities can be combined to account for the total load imposed by each of these Executive functions. For example, the load on the system created by the servicing of a Data Communication Channel input is composed of three Executive activities measured by the statistics gathering system: DCC trap servicing, DCC input data servicing, and a scan of the priority table to start (or restore) the highest priority program which is ready to operate.

Results

During the simulated mission, the CPU statistics were taken for each 30 second interval from approximately $1\frac{1}{2}$ minutes before launch until $19\frac{1}{2}$ minutes after launch. The Executive statistics were taken during two twenty second intervals occurring at approximately 1 minute and 9 minutes after launch. The analysis will concentrate on the data taken in these two sample intervals. Figure 3-3 shows the history of the CPU utilization during the mission. The percentage of total capacity used in Mission and Executive Programs is indicated in the figure. The gathering of Executive statistics imposes an additional load on the system as shown by the two points labled with an S on the figure.

Within the two sample intervals, the system loading has been separated into its several components. Tables 3-3 and 3-4 indicate the CPU loading profiles for the two intervals. For each system activity a percentage of capacity is given. Three parameters of capacity are used: total available capacity, capacity actually used working in Mission and Executive, and capacity used in the Executive alone. In addition to these capacities, the number of occurrences of each of the Executive activities is indicated. For example, Table 3-3 indicates that the total available capacity used working in Mission plus Executive was 57.96%. Of the total available capacity, the Mission programs used 31.32% and Executive used 26.64%. Considering these same activities from the viewpoint of capacity used working in Mission plus Executive (57.96%), the table indicates that roughly 54% was used in Mission and 46% was used in Executive. In the last column the percentage of Executive capacity used by each of the Executive system functions is given. For example, the servicing of 448 DDC inputs required 7.90% of the capacity used by Executive.

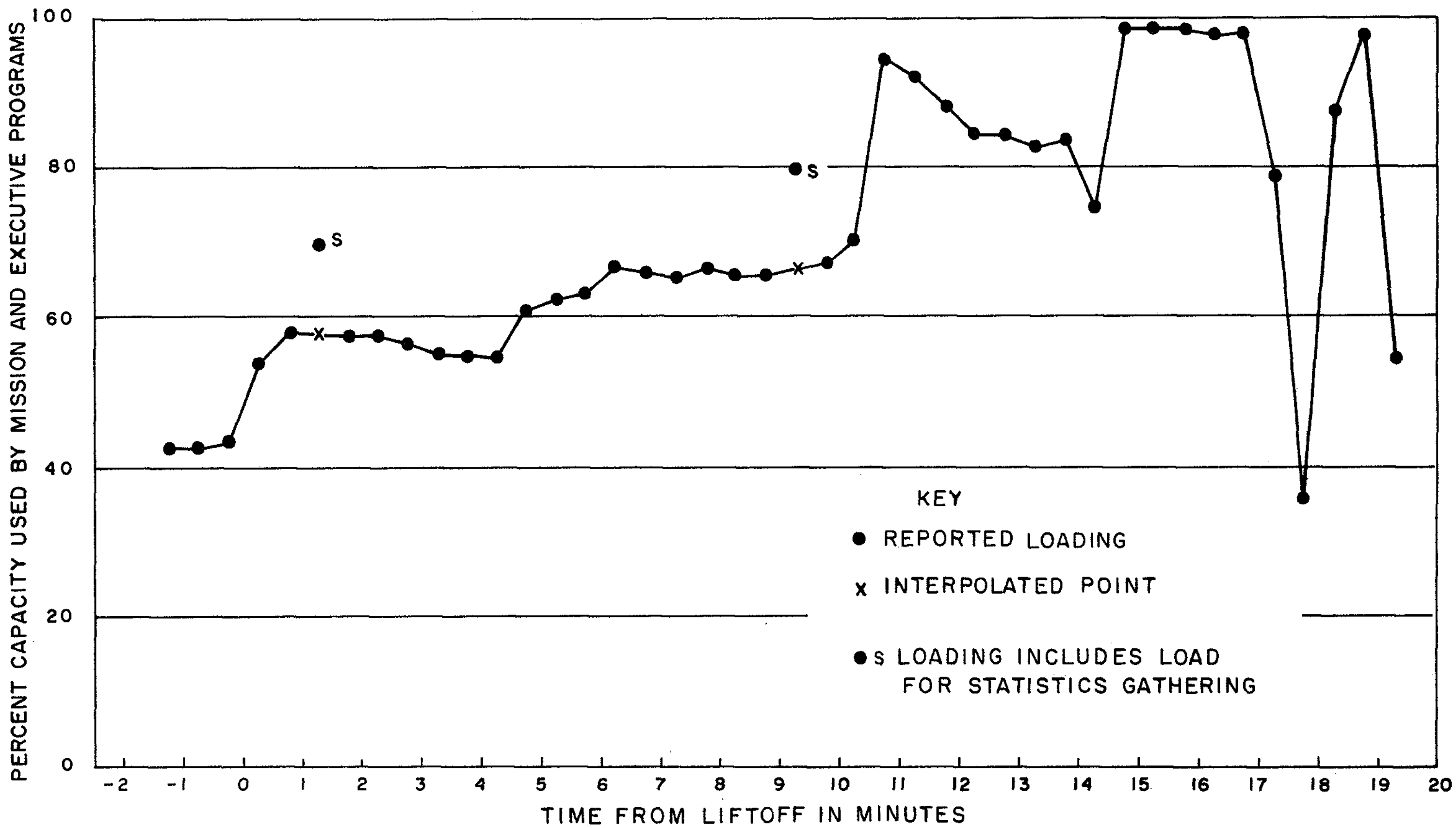


FIGURE 3-3 CENTRAL PROCESSOR UTILIZATION, APOLLO 201 MISSION SIMULATION

TABLE 3-3

APOLLO 201 SIMULATION

CPU LOADING PROFILE AT 1 MINUTE AFTER LAUNCH

System Activity	Number of Occurrences	% of Capacities		
		Total Available	Working	Executive
Statistics Gathering *		18.76		
Idle or Waiting *		23.28		
Working * (Mission & Executive)		57.96		
Mission *		31.32	54.04	
Executive		26.64	45.96	
DCC Inputs	448			7.90
DCC Outputs	48			1.00
D/TV Outputs *	164			3.92
Logging	100			8.60
Linkage	1694			12.91
Misc. Services *	606			5.30
Print Write *	17			1.15
Z Table I/O *	2768			57.30
Core Allocation	3			.35
Core File Allocation	0			0.00
Sequencer Overhead	560			1.58

* These loads are not explicitly given by the statistics gathering system but were derived by allocating the measured system load on the basis of expected system activity. See explanation in text.

TABLE 3-4

APOLLO 201 SIMULATION

CPU LOADING PROFILE AT 9 MINUTES AFTER LAUNCH

System Activity	Number of Occurrences	% of Capacities: Total Available	Working	Executive
Statistics Gathering *		19.58		
Idle or Waiting *		13.58		
Working * (Mission & Executive)		66.84		
Mission *		35.31	52.83	
Executive		31.53	47.17	
DCC Inputs	883			13.64
DCC Outputs	53			1.11
D/TV Outputs *	201			4.83
Logging	113			10.14
Linkage	1736			11.14
Misc. Services *	521			3.96
Print/Write *	101			5.85
Z Table I/O *	2906			47.63
Core Allocation	3			.29
Core File Allocation	0			0.00
Sequencer Overhead	576			1.39

* These loads are not explicitly given by the statistics gathering system but were derived by allocating the measured system load on the basis of expected system activity. See explanation in text.

It should be noted that all of the information required to make allocation of system loads to the various functions is not explicitly given by the statistics gathering system. Items which are asterisked in the table were derived from other system considerations. For example, the percentage of capacity working in Executive and Mission programs was derived by averaging the loads in the intervals prior to and following the Executive statistics sample interval. This work load was subtracted from the total load for the 30 sec. sample interval to obtain CPU load attributable to statistics. Because statistics were taken for only 20 seconds of the 30 second interval, the percent of central processor capacity used in gathering statistics was increased by 50%. This prorated increase in statistics gathering also resulted in a corresponding decrease in capacity spent in idle or waiting.

Discussion of Results

The results indicate for both sample intervals that the Executive work load comprises roughly half of the used system capacity. It should be noted that some of the work load indicated as mission is actually work performed by relocated Executive processors. The capacity used by these programs is measured by the processor statistics gathering system which was not included in the mission simulation run. At any rate the amount of capacity used for these relocated Executive programs is small; during a GT-4 playback mission it amounted to roughly 1.7% of the relocated program work load.

While half of the used system capacity may seem to be a large price to pay for Executive functions, one must be cautious about criticizing a system on this basis alone. The Executive load may be high simply because it performs so many services for the mission programs. If these services had to be performed by the mission programs, the additional computing capacity used by the mission programs might outweigh the savings in Executive. In another case, the ratio of Executive usage to mission usage may be high because the amount of time to perform mission calculations is small. For example, if it takes 600 μ sec for the Executive to transfer from one program to the next and each program only requires 300 μ sec to perform its task, the Executive would use 2/3 of capacity and mission programs would use 1/3 of capacity; this crude example assumes no other activity being performed. An improvement could be sought in this case by consolidation of mission program functions to require fewer linkage calls through the Executive.

There are three areas where possible improvement in Executive loading should be explored.

Z Table Consolidation

Turning once more to the results in Tables 3-3 and 3-4, it can be noted that the bulk of Executive loading, 57% and 47% respectively, is involved with moving Z Tables to and from the Core-File. This traffic load will be discussed to show how consolidation might payoff in a reduction of Executive work load. Mission programs, upon calling for I/O, may request several tables to be moved through the use of multiple arguments in the call. Considering both intervals, the mission programs averaged 1.4 argument sets per call. The Executive however, makes up a separate I/O command for each table, indicated by a single argument set, and must service an individual channel trap for each table transferred. Mission programs frequently require more than one table; some require as many as six tables to be read in during the course of operation. If these tables could be consolidated into one larger table which could be read in as a single block, the Executive load would be reduced. As an alternate solution, if they could be placed adjacent to one another on the core file, and a new table defined which is equal to the sum of these smaller tables, the group of tables could be read in as a single block.

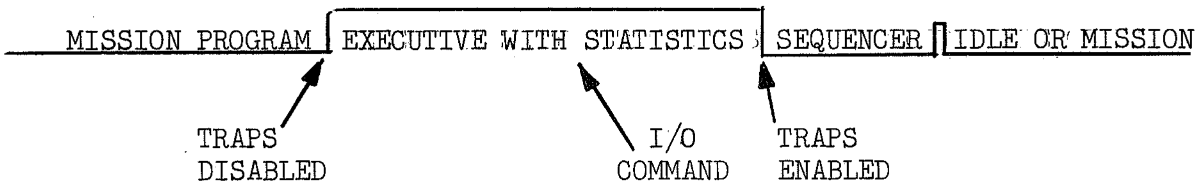
No quantitative estimate of the savings in Executive loading through consolidation of Z Tables has been made because the necessary data for such an analysis was not readily available.

Storage of Z Tables

An analysis of the distribution of traps which indicate the end of a data transmission, shed some light on the amount of time required to accomplish Z Table transmissions to and from the Core-File. A typical interval of system processing is shown in the figure below.

Figure 3-4

Typical Processing Interval



If traps were random with respect to time of arrival one might expect that the frequency of trapping in a mission program would be roughly proportional to mission program duration. Similarly one would expect the number of traps in Executive and in idle time to be proportional to their duration. During the first sample interval there were 3576 I/O traps. These traps were distributed as follows among the three system activities.

Traps during Executive	2860
Traps during Idle Time	418
Traps during Mission Programs	298

A certain number of these traps are expected to occur randomly in time. DCC inputs are expected to be random with respect to cycles of the Executive. Similarly I/O operations involving large blocks of data on relatively slower channels can be expected to exhibit a random distribution with respect to the time when Executive is operated. This latter class includes logging outputs, core allocation I/O, D/TV outputs and DCC outputs. The number of I/O traps associated with these operations during the sample interval was 774. These are all expected to occur randomly. The remaining 2802 I/O operations were associated with transmissions to or from the Core-File. Based on the percentage of time spent in each system activity, one can calculate the number of random traps one would expect to occur during each system activity. This calculation is given in the table below.

TABLE 3-5

Distribution of I/O Traps

System Activity	% of Time	Random Traps Expected	Actual	Core File Surplus
Idle	23.28	180	418	238
Mission	31.32	243	298	55
Executive and Statistics	45.40	351	2860	2509

Subtracting the number of random traps expected from the number actually experienced gives residual Core-File traps. Of these traps, 2509 occur in the Executive Sequencer. This accounts for approximately 89% of all Core-File transmissions. It is interesting to note that these transmissions will occur during Executive Sequencer only if the table is sufficiently short to be read in during the time it would take to complete the Executive Sequencer operation after the channel command had been issued. The timing for these events is approximately as follows:

Issue channel command	$\approx$ 0 μsec.
Record statistics	139 μsec.
Operate Sequencer	152 μsec.
Total Time	$\approx$ 291 μsec.

Thus, roughly 300 μsec is sufficient time to accomplish almost 90% of all Core-File transmissions. This corresponds to the time required to transmit a table of about 75 words. These rough approximations suggest that savings in Executive work load might be achieved by storing the most frequently used short tables in main core and using an in-core-move rather than a Core-File transmission to move the data to or from the mission program. Data presented in the section on Core-File Utilization indicates that 129 Z Tables are shorter than 75 words. The storage required to accommodate these tables is 4030 words. This indicates an average table length of 31 words for this group of tables. The current load borne by the Executive in accomplishing a Core-File transmission and reinstating the program which called for the table is about 1300 μ sec. Using the core-to-core move would require about 800 μ sec. for a 31 word table. Thus if main core was used to store these short tables a saving of about 38% on roughly 90% of the Z Table moves could be effected. Since Z Table moves comprise 57% of the executive work load, a total savings for Executive can be estimated at about 19% of Executive capacity (.38 savings X .90 transmission X .57 load = .19 savings).

These savings can be passed on to effect an increase in mission capacity. The ratio of mission to Executive utilization for the first sample interval was 54% mission to 46% Executive. If the Executive capacity is reduced by 19% this results in a new sharing ratio of 54 to 37. Normalizing this ratio to a base of 100 results in a ration of 59 to 41. Comparing the former mission capacity, 54%, to the latter, 59%, a 9.2% increase in mission capacity is indicated. Similar calculations on the second sample interval result in an increase in mission capacity of about 8.7%. This lesser gain is a reflection of the fact that during the second sample interval the Z Table traffic comprised 47% of the Executive load whereas in the first sample interval it comprised 57% of the Executive load.

Although this analysis is based directly on the Executive statistics, the reader is nevertheless cautioned that several assumptions and approximations indicated above were required which could not be verified by experimentation with the system.

It should be noted that the consolidation of Z Tables, in conjunction with storing the more frequently used Z Tables in main core, may not produce a significant savings over and above placing them in main core alone. The reason for using main core is to take advantage of the fact that the tables are short whereas the advantage of consolidation lies in the fact that through consolidation a longer table is read in but less frequently.

Core Allocation

During both of the sample intervals core allocation imposed very light loading on the system, less than half a percent of Executive load. This was due to the fact that programs had to be allocated only three times in both of the two sample intervals. This in turn is a direct result of the fact that almost all of the programs required during that phase of the mission fit into main core.

Why dwell on so insignificant a load on the system?

One of the principal reasons for going to a 65 K word main core was that with the 32 K main core, the core allocation routine was so busy allocating programs that it took a large portion of system capacity, so large a portion that the system could saturate when the total size of the programs required in a phase of the mission was significantly larger than the area of main core available. In addition to increasing the size of main core, the core allocation routine was modified to reduce the portion of capacity it would require when overload conditions were being reached.

The core allocation routine is a very sophisticated routine which allocates space in main core on the basis of the size and priority of the program to be placed, and the size, priority and status of the programs it may have to displace. The operating time of the core allocation routine is about 5,600 μ sec, and the I/O servicing requires about 560 μ sec. This represents a total loading of 6,160 μ sec to allocate space for a program, move it into core, and enter it.

Data on the average running time of the programs allocated by this system was not recorded during the mission simulation. One can make a rough estimate of the time by assuming that each linkage call represents a new task and, therefore, a new usage of a program, and dividing the amount of time in mission programs by the number of linkage calls. The amount of time spent in mission programs over the two intervals was 13.32 seconds; the number of linkage calls was 3430. This represents 3,697 μ sec of mission program utilization for each linkage call.

This exploration into core allocation has not revealed any concrete problems, it only raises a question. Could a less sophisticated core allocation routine which imposes a less severe load on the system improve the overall loading situation when many programs may be competing for space in main core? The current core allocation routine takes more time than the average time of operation of the relocated programs. Intuitively this appears to be an area for concern.

Summary of System Loading Analysis

Three topics have been treated in study of system loading. In each case, the discussion attempted to point out where savings in executive loading could be achieved. In summary, savings may be achieved by consolidating the Z Tables in such a way as to require fewer I/O calls by mission programs. Savings in Executive operating time may be afforded by storing the Z Tables, which are shorter than 75 words, in main core. A concern was expressed over the possible inefficiency implied by a core allocation routine which requires more time to operate than the programs it moves into main core.

An estimate of increased mission program capacity was made only for the storing of Z Tables in main core. The analysis concluded that a potential 9% increase in mission capacity could be afforded by that change.

Although it is not directly supportable by the results of this analysis, it would appear that the basic system organization was not chosen on the basis of minimizing system loading. It appears rather that the designer sought to achieve a balance between system flexibility and system loading both of which are desirable design goals. The flexibility of the system has been achieved to a certain extent through an increased loading by Executive. For example, the restriction against mission programs directly accessing the data tables causes about half of the load borne by Executive. This restriction is, however, required if the address protection feature is to be used. This restriction is also required if the flexibility of using virtually addressed tables in the mission program is desired. The alternate approach would require that mission programs have direct knowledge of where the tables are stored.

SYSTEM FLEXIBILITY

This section discusses briefly some aspects of flexibility of the Software System Design. In discussing the Software System Design, it will be considered to be synonymous with Executive System Design, since the Executive must be a reflection of the concepts of the System Design.

Sources of Data

The following sources of data have been used:

1. Various documentation relating to design of the Executive and Mission Programs
2. Discussions with James C. Stokes, Assistant Chief, Real-Time Program Development Branch, Mission Planning and Analysis Division
3. Discussions with Richard A. Hoover, Chief of Mission Control Requirements Branch, Flight Control Division

Characteristics of Software Flexibility

The principal characteristics of the software system that represent flexibility are:

1. Ability to add new programs or new functions
2. Ability to change programs
3. Ability to reconfigure the system for a new mission

These characteristics will be discussed as they relate to aspects of the design of the software and hardware systems that influence flexibility.

Impact of Hardware on System Flexibility

The principal aspects of the hardware system that influence flexibility are:

1. Address relocate feature
2. Address and instruction protection feature
3. Size of main core and the core-file

These features have been described in the General Functional Description in Section II. The first two have been key factors in the design of a modular multiprogrammed system. The address relocate feature permits the execution of a program from any part of core without changes to addresses within the program. It is necessary only to set the Relocation Register. The address and instruction protection feature permits the operation of a multiprogrammed system, incorporating many individual programs, controlled by an Executive, without the danger that any important parts of the system will be destroyed by incorrect transfers in a running program. Thus mission - i.e., non-Executive - programs are protected from each other and the Executive is protected from all mission programs. This multiprogrammed capability has not been obtained without penalty. In particular, the address and instruction protection feature denies a mission (relocated and protected) program the right to access memory outside its own area. Thus many service routines of general use, such as I/O routines, may not be used directly, but must be provided, by the Executive, as a service to mission programs. Direct communication between mission programs, such as transfers of control, transfer of data, reading common data or writing in common tables, is also forbidden, and the capability is provided as a service by the Executive. Even if such restrictions were not a result of address and instruction protection, they would be necessitated by the modular multiprogrammed design.

The provision of centralized services by the Executive for I/O, Service Routines and Program Linkage, throws a heavy burden on the Executive and account in part for the high percentage of CPU time taken by it.

A factor which will influence the ability to add new programs to the system is the amount of space available on the Core-File. As long as there is space available this presents no problem, but if space is getting short on the Core-File, and this was beginning to happen for GEMINI 9 as was indicated in the section on Core-File Utilization, the question arises of whether to displace other programs or to use Core-File Allocation from tape. Core-File Allocation has been discussed briefly under Core-File Utilization in this section.

The amount of core storage available for mission programs (bufferable core) has a very important bearing on system flexibility, but the problem has not been analyzed in this study. The problem has been solved by the Executive procedure of Core Allocation, and as far as mission programs are concerned, it is entirely taken care of. However, the effectiveness of core allocation is dependent very much on the amount of bufferable core storage available.

Impact of Software on System Flexibility

The principal aspects of the software system that influence flexibility are:

1. Executive System Design
2. Design of certain individual mission programs

The Executive has no knowledge of the mission except insofar as it has information in a "Routing Table" which may tell it to queue a program on the receipt of certain data. The demand for processing operations is otherwise generated entirely within the mission programs. This means that it is not necessary to modify the Executive in order to insert new mission programs into the system, or in order to change mission logic, unless it is necessary to modify the Routing Table, which has been designed for change, and can be dynamically updated during a mission. The fact that new logic can be incorporated into the system, even to the extent of a complete new mission logic, essentially without changes to fixed core (Executive), is a great advantage to the system.

Although the mission logic and the development of the sequence of processing of mission programs is almost entirely out of the control of Executive, once any part of the sequence is formulated in a mission program and the need for operation of another mission program has been determined, it is left to the Executive to handle the mechanics of finding space in core for it (core allocation), bringing the program into core, supply the data for it and causing it to operate. Thus the programmer is absolved from the responsibility of doing many routine tasks, including handling linkage between programs and finding space for a new program. This aspect of finding space in main core for programs is very important and the reduction of this problem to a routine process of core allocation, out of the realm of the mission programs, contributes greatly to the flexibility of the system.

The Software System has been designed in a modular way and this provides a certain basic flexibility. It is therefore easy to insert a new program. However, it may be pointed out that inserting a new program does not cause the program to be used. Some other changes would have to be made, usually to a supervisor to have it request the Executive to queue the new routine.

The ability to change a program is greatly facilitated by the modular design of the software system. The old program is removed and the new program, interfacing in the same way with other mission programs, is put in. However, the ability to change a program must be considered to be primarily a question of the design of the particular program, and this may determine whether the changes are easily incorporated or whether a complete re-write is required. It may also be related to the interface between this program and the outside world (through the Data Communications Channel). A good example of this is the programming required to drive the Digital Displays. The routing of data to individual lights is controlled jointly by the Digital Display Driver Processor, which receives data from mission programs, and by a form of patchboard wiring. In order to reconfigure the digital displays, changes are required both in the wiring and in the programs. It would be possible to design a program which would set up digital displays in a completely general way, relying only on an internal table to route data. Patchboard wiring could be frozen and changes to the displays could usually be taken care of by changes to the table. However, at the time the MCCH was designed, it was considered by NASA and IBM that this would throw too much of a load onto the IBM 7094. Hence the system was set up to be partly dependent on the hardware. The situation is not flexible, but it should be understood that the lack of flexibility is due, in large measure, to the design of the Digital Display Driver Processor (which happens to be part of Executive), and that this design is not a reflection of an inflexible Executive.

The modularity of design is carried into the design of data tables. Instead of a large fixed pool of data to which programs can refer, the data is broken up into tables, most of which are stored on the Core-File, and they are made available as required by the program. It can be pointed out that there are other reasons for adopting such a modular approach to data tables. There are over 124,000 words of storage required for data and clearly this could not be stored entirely in main core. Furthermore the address and instruction protect feature prohibits mission programs from referring directly to any data stored in such a data pool.

<u>Summary</u>

This section has dicussed, in a general way, certain aspects of the software system flexibility. It was pointed out that certain features of the hardware and software systems contribute to software flexibility. These include the address relocation feature, the address and instruction protection feature, the modular design of programs and data tables and the provision of centralized services by the Executive. On the other hand it was shown that the address and instruction protection feature imposes certain restrictions on the system. Other features have been discussed in order to indicate, in a general way, how they can affect flexibility.

No particular conclusions have been reached. Such conclusions would, in any event, be somewhat subjective in nature.

SECTION V

SUMMARY OF SIGNIFICANT CONCLUSIONS

CORE-FILE UTILIZATION

The analysis has considered the problem of storage on the Core-File, and methods of alleviating a shortage of space have been discussed. Core-File storage can be saved by allocating routines from magnetic tape to the Core-File by phase of the mission, or by removing selected routines from the Core-File and allocating these routines from tape as required. Both of these alternatives can be used if desired. It is felt that the second alternative, to remove selected routines from the Core-File and to allocate these routines from tape, would provide a satisfactory solution incurring no processing delays, or, at worst, acceptable processing delays, if the routines were carefully selected.

SYSTEM LOADING

The analysis of Executive loading revealed three areas for possible improvement in the system. Savings in system loading might be achieved by consolidation of the Z Tables stored on the Core-File, by storing the most frequently used short Z Tables in main core, and by modifications to the core allocation routine to shorten its running time. A 9% increase in mission program capacity is estimated for the second change; no estimate of increased mission program capacity was made for the other two changes.

Much of the Executive load is attributable to the achievement of a flexible system design. As much as half of the load is due to the fact that programs are not permitted direct access to the data tables. In a more rigid system this cost in system loading might be reduced substantially, but this would probably result in a higher cost for development and testing under the more rigid system design. In a less rapidly evolving system this might be advisable; for the Mission Control Center application the flexibility of the current system is used to advantage in preparing for the mission-to-mission changes, and a more rigid system might cost more in production time and dollars than would be justified by a savings in system loading.

SECTION VI

SUMMARY OF RECOMMENDATIONS

The study of the RTCC has highlighted the kinds of questions that are of interest. The study should now turn to the new IBM 360-75 System. Some of the areas of interest will be very similar to corresponding areas in the IBM 7094 System. However, due to the different characteristics of the IBM 360-75 System, it is clear that some problem areas that were quite critical on the IBM 7094 will just disappear on the IBM 360-75, or will appear in a vastly different form.

The study of the IBM 360-75 System should follow the same salient objectives that were suggested in this report.

1. Capacity and Capability of the System to Meet Requirements
2. Effectiveness With Which the System Meets These Requirements
3. Ability of the System to Respond to Changing Requirements

The performance of the RTCC System has not been related directly to Mission Support Requirements. Furthermore, the study came at a time when this computer system was on the point of being phased out, to be replaced by the IBM 360-75 system. Therefore it would be meaningless, for example, to talk of the ability of the system to support new programs, such as future APOLLO missions. There is no possibility of this kind of requirement being levied on the IBM 7094 system. The ability of the RTCC to respond to new, and as yet undefined, requirements will therefore be a significant new study area for the IBM 360-75 system. The study should consider mission support requirements for future APOLLO missions, and analyze the ability of the IBM 360-75 system to meet these requirements. The study will further analyze the capability of the system for growth - growth in hardware system configurations, growth in software, and the ability of the system to respond to growing requirements.

The Real Time Operating System (RTOS) for the IBM 360-75, which corresponds to the Executive on the IBM 7094, presents a very interesting study area from two points of view. In the first place the 360/OS was signed as a general purpose operating system for the IBM 360-75, and it is being tailored to the needs and requirements of the RTCC. The question arises immediately: "How well will the RTOS perform the job for the RTCC and will it perform adequately in all respects?" There is little doubt that, as time goes on, experience will indicate that certain modifications will be required. The second point is that the RTOS has been designed as a multi-level multi-programmed system. Mission Control could be just one job of many that are operating simultaneously on the computer. There is no doubt that this will not be attempted very soon. However the possibility exists for the future, and considerable study and testing will certainly be required before such a mode of operation would be accepted.

Another question of great interest is the capability and capacity of the computer system to do the job within the time constraints of the real-time environment. This question has not been considered so far in the study. A satisfactory answer cannot be given in terms such as "Yes, the system can meet the time constraints of the real-time environment" or "No, it cannot ...." The answer would almost inevitably be the former, because mission support requirements are adjusted to the point where they will be met by the RTCC. For the future, it is expected that the IBM 360-75 system will meet mission support requirements, with spare capacity, for some time to come. It is therefore important to discuss such questions as "How effectively does the RTCC system meet mission requirements?" and "How efficiently does the RTCC software system work?"

Mission Programs have not been studied in any depth so far in the study. However, some of the characteristics of system loading in the Executive can be attributed directly to the way things are done in Mission Programs. For example, the use of Z Tables by Mission Programs has probably not been optimum in the present system, as was suggested by the analysis of program statistics. Some questions of flexibility are also dependent on Mission Program design. These things and many others should be considered in more detail.

In order to facilitate some of the studies of system characteristics and loading, it is suggested that the taking of program statistics could be tailored more closely to the objectives of the studies. Furthermore it is considered that the load on the CPU could be lightened by providing options on the taking of statistics. Then only those statistics that are really required would be taken. For example, the timing of sections of Executive takes up the bulk of the time used by statistics gathering. If it were an option to take counts of the number of times that sections of the Executive operate, with or without timing, the load could then be reduced by taking counts alone. Previous runs would have established good average times for these sections of the Executive. It if were possible to reduce the load for statistics gathering to a small fraction of the present load, it might become more acceptable to take these statistics during simulations, and even during a mission.

APPENDIX A

This appendix contains three tables. They are:

1. Table A-1 Core-File Storage Requirements by Mission Phase
2. Table A-2 Z Tables: Number of Tables by Size of Table
3. Table A-3 IBM-7094-II Equipment List

TABLE A-1 CORE-FILE STORAGE REQUIREMENTS BY MISSION PHASE

PROGRAM NAME	PROGRAM DESCRIPTION	USAGE CODE	MISSION PHASES ALL	P1	L1	P2	L2	O	R	L2+R
	<u>Executive Programs</u>:									
1. GAEXEC	Executive Monitor (in-core)									
2. IXDISP	Executive Display Processor		1724							
3. XPPTDP	On-Line Print Program		1748	34						
4. XPTYPE	Teletype Processor		1510	49						
5. XSCFLN	Core File Allocation Supervisor (on tape)									
6. XSFIXR	Not Ready I/O Error Supervisor		286	51						
7. XSREST	Executive Restart Processor		746	58						
8. XSSWVR	Switchover Supervisor		386	61						
9. XSTATC	Gather Statistics during Real-Time Run		748	68						
10. XSTAT1	CPU Utilization Supervisor		736	75						
11. XSTAT2	Processor Timing in Real-Time		1308	88						
12. XSTAT3	Executive Logic Timing in Real-Time		1022	98						
13. XSTPCF	Tape-Core File Transmission Processor (on tape)									
14. XXDDDI	Digital Display Processor		674	104						
15. XXMEDX	Executive MED Processor		778	111						
16. XXMESS	On-Line Error Program		566	116						
17. XXSYNP	Stop Card Supervisor		52							
18. XXTAPE	Real-Time System Tape Reader		702	123						
				130K						
	<u>Mission Programs</u>:									
	A. <u>Supervisors</u>									
1. MSADCS	DCS Supervisor	F	1620							
2. MSALAH	Agena Launch Supervisor	F			2292					
3. MSCTRL	Mission Planning Control Supervisor	F				546		546		
4. MSDATE	Special Display Supervisor	F	812							
5. MSDCUP	Orbit Differential Correction	FT				5150		5150		

PROGRAM NAME	PROGRAM DESCRIPTION	USAGE CODE	MISSION PHASES							
			ALL	P1	L1	P2	L2	O	R	L2+R
6. MSDKIT	Mission Planning Supervisor	F				2702		2702		
7. MSDPCS	Orbit Display Supervisor	F				762		762		
8. MSDRAW	Projection Plotter Supervisor	F	4070							
9. MSGABT	Gemini Abort Supervisor	C					2200			2200
10. MSGLAH	Gemini Launch Supervisor	F					3796			3796
11. MSGMNI	Gemini Mission NI Supervisor	F				2248		2248	2248	2248
12. MSGPMB	General Purpose Maneuver Supervisor	I				1148		1148		
13. MSGREN	Reentry Supervisor	F						7594	7594	7594
14. MSLSIN	L. S. Input Supervisor	FT	2218							
15. MSMANC	Manuever Control Supervisor	FT				8242		8242		
16. MSNLED	Nominal Launch & Ephemeris Supervisor	F				2894				
17. MSTBTI	Two Impulse Solution Supervisor	FT				2538		2538		
18. MSTIME	Orbit DC & Trajectory Time Supervisor	F				1730		1730		
19. MSTLMS	Telemetry Supervisor	F	2254							
20. MSTRAJ	Orbit Supervisor	F				1244		1244		
21. MSTTFR	Time-to-Fire Supervisor	FT				3252		3252		
22. MSUPER	Mission Supervisor	F	4304							
	B. <u>Processors</u>									
1. MYTEST	Temporary Test Processor	I				32		32		
2. MMABCR	Agena DCS On-Board Reset Clock Processor	IT				928		928		
3. MMACPT	Agena DCS Command Load Change Processor	IT				484		484		
4. MMAECG	Agena Digital Command and Ephemeris Processor	IT				1000		1000		
5. MMAMCP	Agena DCS Memory Compare Processor	IT				142		142		
6. MMANUL	Mission Manual Input Processor	I	2344							
7. MMAPCL	Agena DCS Pad Load Update Processor	I		132						

PROGRAM NAME	PROGRAM DESCRIPTION	USAGE CODE	MISSION PHASES ALL	P1	L1	P2	L2	O	R	L2+R
8. MMAQCL	Agena Data Quality Calculation Processor	F			504					
9. MMBARB	Bank Angle - Bank Angle Reverse Processor	IT						2120	2120	2120
10. MMBCNV	DC Convergence Processor	IT				2504		2504	2504	2504
11. MMBFED	DC Pre-Edit Processor	IT				826		826	826	826
12. MMBMDF	DC Weight Modification Proc.	IT				404		404	404	404
13. MMBNII	Launch Burnout Processor	I					1942			1942
14. MMBRPD	Analytic Ephemeris Generator	IT				4612		4612		
15. MMBSEL	DC Selection Processor	IT				632		632	632	632
16. MMBTDP	Orbital Elements Processor	F				3322		3322		
17. MMGGNG	Agena GO/NO-GO Calculation	IT			958					
18. MMCHAP	L.S. Chaining Processor	IT				1358		1358	1358	1358
19. MMCLNI	Cowell Integrator	IT				3114		3114		
20. MMCNVL	Constraints Evaluation	IT				3458		3458		
21. MMCTTF	Orbit Time-to-Fire Processor	IT						160		
22. MMDGPR	Pre-Retro Indicator Processor	I						914		
23. MMDIMP	Minimum-Maximum Impact Point	F					3190			3190
24. MMDNSC	Day/Night & Station Contacts	I				3932		3932		
25. MMDQCL	Data Quality Calculations	F			464		464			464
26. MMDRUG	Rendezvous Initiator Proc.	I						1344		
27. MMDSMT	Summary Maneuver Table for S/V Missions	I				652		652		
28. MMDYNM	Orbital Dynamic Display Proc.	I				1674		1674		
29. MMEACC	Store Cape Crossing Times	IT				1620		1620		
30. MMEDLA	GLV Nominal Launch Targeting Processor	F				1203				
31. MMELED	L.S. Raw Radar Edit	IT				3224		3224		
32. MMEWST	Store 22 Orbit Ephemeris	IT				1071		1071		
33. MMGDNU	Orbit Navigation Initiator Proc.	IT				1190		1190		
34. MMGMIL	Gemini Iteration Loop	IT				4946		4946		
35. MMGNGC	GO/NO-GO Calculation Proc.	IT					1708			1708
36. MMHMAN	History Maneuver Insert	I				1742		1742		
37. MMIIMP	Immediate Impact Point Calculation	F						1138		

PROGRAM NAME	PROGRAM DESCRIPTION	USAGE CODE	MISSION PHASES ALL	P1	L1	P2	L2	O	R	L2+R
38. MMIVIR	Compute Burnout Vector from IVI ΔV Vector	I							788	788
39. MMLAMX	Agena Display Calculation Proc.	I				748		748		
40. MMLCAD	Advance Maneuver Line and DC Vector	IT				2594		2594		
41. MMLFIT	Launch Curve Fit Processor for ΔV	F			338		338			338
42. MMLGGS	Gemini/Agena Scheduling Proc.	I				3188		3188		
43. MMLINT	Launch Initialization Proc.	I		712		712				
44. MMLTTF	Launch Time-to-Fire Proc.	I					1346			1346
45. MMMAIL	Agena Iteration Loop	IT				5924		5924		
46. MMMFUD	Freeze, Unfreeze, Delete	I				3344		3344		
47. MMMISS	Missing Data Processor	I			146		146			146
48. MMMLUR	Maneuver Line Relocation	IT				2436		2436		
49. MMMSTA	Station Characteristics	I	832							
50. MMMUGD	Maneuver Initiator Processor	IT				498		498		
51. MMNATB	NAG/NA Table Generator w/o NADAL Crossing	IT				2624		2624		
52. MMPAMC	Agena DCS Maneuver Proc.	IT				1840		1840		
53. MMPHLG	Phase Lag Computations at M-1	IT				3310		3310		
54. MMRBRC	Primary Time-to-Fire Proc.	IT						1496		
55. MMRDLS	DC Residuals Display Proc.	I				3358		3358		
56. MMREDP	Reentry Display Processor	F						976	976	976
57. MMRMTA	Remote Site Acquisition Data	IT				2538		2538	2538	2538
58. MMRRED	H.S. Raw Radar Edit Proc.	F			1966		1966			1966
59. MMRSMD	H.S. Raw Radar 2n+1	F			780		780			780
60. MMRXNI	Orbit Runge-Kutta Integrator	IT				2380		2380		
61. MMRYNI	Reentry Runge-Kutta Integrator	I							3580	3580
62. MMSADC	H.S. Raw Radar Short Arc and Data Conversion Proc.	F			2620		2620			2620
63. MMSGPM	GPM Computation Processor	I				3407		3407		
64. MMSMTA	SMY Maneuver Table	IT				2450		2450		
65. MMSTAG	Station Contact Processor	IT				2314		2314		
66. MMTASL	Area Selection Processor	I					911			911

PROGRAM NAME	PROGRAM DESCRIPTION	USAGE CODE	MISSION PHASES ALL	P1	L1	P2	L2	O	R	L2+R
67. MMTDNI	Impact Predictor Integrator	F					1704			1704
68. MMTIAN	Gemini Launch Vehicle Proc.	F		6058	6058	6058	6058			6058
69. MMTLEP	Events Panel Processor	I	1300							
70. MMTPHZ	Terminal Phase Processor	I						4486		
71. MMTSTR	GMTLO Computations	I				865				
72. MMTVSA	Target Vehicle Sight Acquisition	I				1972		1972		
73. MMUDAT	Update Target & Landing Site Tables	I				454		454		
74. MMVGMA	V and Gamma Processor	F			1081		1081			1081
75. MMWTFR	Weight and Fuel Remaining	IT				1128		1128		
76. MMXMIS	Agena Mission Data Processor	I			191					
77. MMYIDG	Bayes Inversion & Delta Gamma	IT				1952		1952		
78. MMYMAN	Bayes Orbit Propagation Proc.	IT				1922		1922		
79. MMYNPS	Numeric Partials for Maneuver	IT				1230		1230		
80. MMYREN	Bayes DC Reentry Propagation Processor	I							1824	1824
81. MMZPLT	Recovery Zone Plot Processor	I				2484		2484		
82. MMZYZZ	Two Impulse Single Solution Proc.	I				3360		3360		
83. MMZZZZ	Two Impulse Driver & Finite Burn Processor	I				4050		4050		
84. MQAGTM	Agena H.S. TLM Input Proc.	F	692							
85. MQAMAN	Agena MED Decoder	I	2118							
86. MQATLS	Agena L.S. Input Processor	IT					1868	1868	1868	1868
87. MQBDHS	Bermuda H.S. Data Input	F			424		424			424
88. MQDCDE	MED Decoder Processor	I	2604							
89. MQGEBD	GE/B Data Input Processor	F					922			922
90. MQGMTM	Gemini H.S. TLM Input Proc.	FT	3168							
91. MQIPSM	IP Smooth Data Input Proc.	F			824		824			824
92. MQMEPT	Telemetry MED Processor	I	2274							
93. MQMMEP	Mission Manual Entry Processor	I	3330							
94. QMPEEP	Manual Input Processor	I	2582							
95. MQRRIP	IP Raw Radar Input Processor	F			590		590			590
96. MQSMEK	Summary Message Generator Proc.	IT	582							
97. MQTMLS	Low Speed Input Processor	IT	4444							
98. MQTNTM	Titan H.S. TLM Input Processor	F		342	342	342	342			342
99. MXAACQ	Target of Closest Approach	I				2846		2846		

PROGRAM NAME	PROGRAM DESCRIPTION	USAGE CODE	MISSION PHASES ALL	P1	L1	P2	L2	O	R	L2+R
100. MXADFK	Agena TLM FDK Processor	I	438							
101. MXAFMT	Agena Launch Display Format Processor	F			1406					
102. MXAHST	Agena H.S. Display Output Proc.	FT	1314							
103. MXANLD	Agena Launch Display Output Proc.	F			224					
104. MXARTT	Agena/Gemini DCS MED Decoding Processor	I	3642							
105. MXASND	Rev. of Ascending Node Display	I				1962		1962		
106. MXATCP	Agena DCS Transmission Proc.	I		1266		1266	1266	1266	1266	1266
107. MXBUSH	Reentry Heating Evaluation from Apogee and Perigee	I				2640		2640		
108. MXCGYA	Agena Continuous Summary	I	1568							
109. MXCNVL	Constraints Violated Display	IT				1926		1926		
110. MXCRUP	Rendezvous Display Processor	F				926		926		
111. MXDCPT	DC Reentry Print Processor	I							568	568
112. MXDDRR	Recovery Room Time Digital Disp.	I				762		762		
113. MXDGTI	Two Impulse Digital Display Proc.	IT				926		926		
114. MXDKIA	DKI Display	IT				1216		1216		
115. MXDMTB	Detailed Maneuver Display	IT				3366		3366		
116. MXDSUM	Differential Correc. Summary Disp.	IT				3404		3404	3404	3404
117. MXDTPH	Terminal Phase Digitals	I				1162		1162		
118. MXEDLA	Gemini Launch Vehicle Targeting Display	I				920				
119. MXFDRD	FDO Reentry Digitals Display	F							1214	1214
120. MXFLAD	FDO Abort Digital Display Proc.	F					678			678
121. MXGADD	Gemini Abort DDD Processor	C					284			284
122. MXGFDK	Forced Display Keyboard Output	IT	454							
123. MXGHST	Gemini H.S. TLM Output Proc.	FT	2370							
124. MXGIDE	TV Guide Formatting Proc.	I	688							
125. MXGRAF	Trend Display Formatting Proc.	I	710							
126. MXGTAB	TAB Display Formatting Proc.	IT	1268							
127. MXHVSV	H vs. V Display to D/TV System	F							662	662
128. MXJACD	Agena DCS Caution Display Proc.	IT	982							
129. MXJCMD	Agena DCS Command Load Display Processor	IT	1356							

PROGRAM NAME	PROGRAM DESCRIPTION	USAGE CODE	MISSION PHASES ALL	P1	L1	P2	L2	O	R	L2+R
130. MXJERR	Agena DCS Error Display Proc.	I	706							
131. MXJKST	Agena DCS Program Constant Display Processor	I	722							
132. MXJMEM	Agena Memory Compare Display Processor	IT	426							
133. MXJRPR	Print Processor for MSCTRL and MSTIME	IT	356							
134. MXJXMT	Agena DCS Octal Command Load Display	I	598							
135. MXLADP	Launch Output Processor	F			2500		2500			2500
136. MXLCDT	Low Speed TLM Display Proc.	IT	892							
137. MXLGAD	LS Log Display Processor	I	740							
138. MXLIMT	Gemini TLM Limits Display	I	808							
139. MXLSDT	L.S. TLM Output Processor	I	1458							
140. MXLZAP	Landing Site Area Table Display	I	954							
141. MXMDIU	Telemetry MDIU Display	I	1016							
142. MXMLMN	Maneuver Line Monitor Display	I				880		880		
143. MXMMDF	Mission Monitor Display	I				866		866		
144. MXMNUP	Maneuver Display Processor	I				1110		1110		
145. MXMSPT	Mission Supervisor Print Proc.	I	682							
146. MXNAUP	DCS Navigation Update Proc.	I				1060		1060		
147. MXNCPC	DCS Network and Command Proc.	I	2126							
148. MXNSTC	Next Station Contacts Proc.	IT				1030		1030		
149. MXPCFS	Calibration Input Processor	I	250							
150. MXPPFP	10 x 10 Footprint Plotter	F						530	530	530
151. MXPRUP	Pre-Retro Updating Processor	I						1174		
152. MXPTML	Time of Longitude Crossing Print	I				1306		1306		
153. MXPURP	General Purpose Maneuver Display	I				872		872		
154. MXQEDP	Ephemeris Data Transmissions	I				480				
155. MXQLDS	Launch Display Output Processor	F			244		244			244
156. MXRADD	Retro Low Abort Digitals Display Processor	C					1622			1622
157. MXRBRN	OAMS Display	I	1076							
158. MXRDIG	Retro Reentry Digitals Display	F						1096	1096	1096
159. MXREDT	Reentry Retro Digitals No. 2	F						982	982	982

PROGRAM NAME	PROGRAM DESCRIPTION	USAGE CODE	MISSION PHASES ALL	P1	L1	P2	L2	O	R	L2+R
160. MXRFIR	Retro Fire Table Display Proc.	IT						1276		
161. MXRLTM	Relative Time Accumulators	IT	322							
162. MXRRWM	Recovery Room Groundtrack	F				1530				
163. MXRYLB	Abort/Reentry World Map Proc.	F					528		528	528
164. MXRZRB	Abort/Reentry H vs. A Plot	F					460		460	460
165. MXSITE	Reentry Station Contacts Proc.	I						830	830	830
166. MXSMTA	Summary Maneuver Display Formatter	IT				1272		1272		
167. MXSSMM	Sunrise/Sunset-Moonrise/Moonset	IT				4012		4012		
168. MXSTIC	Station Contact Table Output	I				2872		2872		
169. MXSUPT	Midcourse Print Processor	I						708		
170. MXTABA	Agena Summary Tab Display	I	1382							
171. MXTBIT	Multiple Solution Display for Two Impulse	IT				1068		1068		
172. MXTDMT	Single Solution Display for Two Impulse	IT				1274		1274		
173. MXTGTB	Target Table Display Proc. (Dummy)	I				880		880		
174. MXTHST	Titan H.S. Output Processor	F		1416	1416	1416	1416			1416
175. MXTIPL	Two Impulse Display	IT				668		668		
176. MXTNDA	Agena TLM Trends	I	948							
177. MXTPCP	Hardcopy TLM Processor	I	340							
178. MXTRAJ	Orbit Output Processor	F				632		632		
179. MXTSUM	TLM Summary Rebr.	IT	2070							
180. MXVECT	On-Line Vector Print Proc.	I	2894							
181. MXVGAM	V vs. Gamma Processor	F			246		246			246
182. MXVTRA	Transmit ACR Vector	IT		538		538		538	538	538
183. MXWMAP	MOCR World Map Orbit Groundtract Processor	IT				1552		1552		
184. MXWZPP	World Map Present Position Proc.	F				496		496		
185. MXXXXX	Pre-Set Position on Reentry Footprint Projection Proc.	I						286	286	286

PROGRAM NAME	PROGRAM DESCRIPTION	USAGE CODE	MISSION PHASES ALL	P1	L1	P2	L2	O	R	L2+R
186. MXYAFP	Reentry/Abort Footprint for X-Y Processor	I					498		498	498
187. MXYBPP	Reentry/Abort Present Position for X-Y Phi Lambda Proc.	F					326		326	326
188. MXYCBT	X-Y Orbit Display Proc.	F				2998		2998		
189. MXYDIIL	X-Y H vs. Lambda Digitals Proc.	F				452		452		
190. MXYEHL	X-Y vs. Lambda Present Position Processor	F				326		326		
191. MXYFLB	Reentry/Abort Present Position	F					246		246	246
192. MXZPLT	Recovery Zone Plot	I				534		534		
193. MXZREC	Recovery Zone D/TV Digital Display	I				3068		3068		
194. NSUPER	ORACT Supervisor		402							
	SUBTOTALS		92,104	10,464	25,614	202,522	45,534	215,080	42,694	83,036
	PROGRAMS IN ALL PHASES			92,104	92,104	92,104	92,104	92,104	92,104	92,104
	Z TABLES			124,501	124,501	124,501	124,501	124,501	124,501	124,501
	TOTAL IN EACH PHASE			227,069	242,219	419,127	262,139	431,685	259,299	299,641

TABLE A-2 Z TABLES: NUMBER OF TABLES BY SIZE OF TABLE

Size	No.	Size	No.	Size	No.	Size	No.	Size	No.
1	2	42	1	132	1	264	3	736	1
2	3	45	2	140	1	287	2	757	1
4	1	47	1	145	1	300	2	800	1
5	1	50	6	150	1	340	1	804	1
6	5	55	3	154	1	350	3	904	2
7	1	60	15	160	1	360	1	999	1
8	3	61	1	162	1	364	2	1000	1
9	3	70	2	165	1	385	2	1020	1
10	3	75	1	168	1	400	1	1110	1
12	5	76	1	172	1	405	1	1200	1
14	3	80	2	175	1	410	1	1344	1
15	5	83	1	180	2	432	1	1456	1
16	2	84	1	184	1	500	6	1540	1
17	2	85	1	188	1	512	1	1584	1
18	1	90	4	192	1	528	2	1628	1
19	1	92	1	199	2	532	1	1680	1
20	9	94	4	200	5	550	1	1710	1
21	1	95	2	210	2	581	1	2000	2
25	4	96	1	220	1	600	3	2047	1
26	1	100	4	222	1	630	1	2064	1
30	12	102	1	226	1	673	1	3000	1
31	2	106	1	232	1	678	1	3100	1
34	1	112	1	242	1	683	1	4824	2
35	3	120	2	244	6	700	4	4950	1
36	3	122	1	245	1	707	1	5000	1
39	1	125	1	250	2	728	1	8505	1
40	19	129	1	261	4	730	1	9620	1
								11400	1

TABLE A-3 IBM 7094-II EQUIPMENT LIST

The following list comprises the principal standard components and the special equipment of each IBM 7094-II.

TYPE AND MODEL	NAME	QUANTITY
7111	Instruction Processing Unit	1
7109	Arithmetic Sequence Unit	1
7606-3	Multiplexor	1
7302-3	Core-Storage	1
7151-2	Console Control Unit	1
7608	Power Converter	1
7618	Power Control	1
7607-III	Data Channel	1
7607-IV	Data Channel	1
7617	Data Channel Console	2
711-II	Card Reader	1
716	Printer	1
729-VI	Magnetic Tape Unit	12
2361-A	Large Capacity Storage, Model 2	1
7281-II	Data Communication Channel	1
7286-V	Direct Data Channel	1

APPENDIX B

This Appendix contains a bibliography of documents consulted in accomplishing the analysis reported in this Part I of Volume III.

1. IBM RTCC Systems Design Workbook.
2. RTCC Programmer Working Book, Vols. 1, 3, 4, 5, 6, 7, and 12.
3. Notes on The RTCC Real-Time System (IBM), Stanley & Jodeit, Gemini Operational Program Functional Specifications.
4. System Performance Memoranda (Informal IBM Reports).
5. RTCC Development Plan, 5 Nov. 1965.
6. IBM 7094 Principles of Operation.
7. IBM 7281 II Data Communication Channel, Customer Engineering Instruction Manual.
8. Ninth Gemini Mission Development Plan, 3 May 1966.

PART II

SIMULATION CHECKOUT AND TRAINING SYSTEM (SCATS)

SECTION I

INTRODUCTION

GENERAL

An intensive study of the Simulation Checkout and Training System (SCATS), its modes of operation and its application to simulated exercises was completed during the past four months. An initial product of this study was a series of drawings defining the general data flow in all the modes of operation and a detailed data flow within three of the four subsystems of SCATS (the Simulation Interface Subsystem was not drawn). The following drawings are being reproduced and delivered to NASA under separate cover:

1. General Data Flow of the Simulated Remote Sites (SRS) Open and Closed Loop Configurations,
2. General Data Flow of the MOCR Open and Closed Loop Configurations,
3. General Data Flow of the Integrated Closed Loop Configuration
4. SRS Detailed Data Flow
5. Data Inputs to SRS Console Modules
6. Data Inputs to Simulation Control Subsystem (SCS) Console Modules
7. Simulation Data Subsystem (SDS) Data Flow

The equipments which make up the four SCATS subsystems are in most cases similar to those equipments which are used by the operational systems in the MCCH or the MSFN. Exceptions are mainly SCATS interface equipments and certain SCATS peculiar equipments used for simulation control.

Equipments similar to those found in other systems are a) the auxiliary display and control, which interfaces the SCA consoles with the Ground Support Simulation Computer (GSSC), b) the GSSC which is an IBM 7094 (part of the RTCC) used for generation of telemetry data, tracking data, sequencing of simulated remote sites, display generation for the SCA and also contains the math models of the various spacecraft and launch vehicles, c) the SRS which contains essentially the same equipments, necessary for flight controller use, that are found at the remote sites of the MSFN, and d) the display and control consoles in the SCA, which permit monitoring of the Simulation Operations Computer (SOC) and the GSSC. The programming and the detailed use of some of these equipments necessarily differ from those in the operational system, but the basic equipments remain unchanged.

Some of the equipments which are SCATS peculiar are a) modules located on SCA consoles, which allow fault insertion, control of simulation, and the monitoring of mission progress, b) simulation adapters attached to the SCATS PCMGS in the SRSS, which control noise and interrupts into the incoming TLM bit stream, and c) the exchange control logic (ECL) and the control status logic (CSL) which are interfaces between the Process Control Unit (PCU) and the GSSC and SCA consoles respectively.

The many modes of the SCATS permit the system to be configured to meet the requirements of the individual phase of the mission to be simulated. Data sources vary for each mode and in some cases are interchangeable to provide backup inputs or to allow equipments, such as the Gemini Mission Simulator (GMS), to be used for other purposes while a SCATS exercise is being conducted. TABLE 1 illustrates the planned modes of operation of the SCATS during the Gemini VIII series of simulated exercises. In particular, it indicates the various data sources planned for the different SCATS modes.

TABLE 1 TYPICAL SCATS MISSION CONFIGURATIONS

EXERCISE	MISSION PHASES	SCATS MODE	OPERATING AREAS EXERCISED	TLM DATA SOURCES VEHICLE	TLM DATA SOURCES SOURCE	TLM DATA SOURCES PHASE	TRACK/TRAJ. DATA SOURCES VEHICLE	TRACK/TRAJ. DATA SOURCES SOURCE	TRACK/TRAJ. DATA SOURCES PHASE	GMCF DATA SOURCE	S/C COMMAND RESPONSE GEMINI	S/C COMMAND RESPONSE AGENA
Simulated Network Simulation	All(1)	Integrated Closed Loop	MOCR/SSR, SRS	Gemini Agena	GMS GSSC	All All	Gemini Agena	GSSC GSSC	All All	GSSC	GMS	GSSC
Network Simulation	All(1)	MOCR-Closed Loop	MOCR/SSR	Gemini Gemini Agena Agena	CKN(2) MSFN(4) GSSC MSFN(4)	Launch(3) Orbit(5) Launch(3) Orbit(5)	Gemini Agena Agena	GSSC GSSC MSFN(6)	All Launch(3) Orbit(5)	CKN	GMS	GSSC
Reentry Simulation	Orbit (final rev.) Reentry	Integrated Closed Loop	MOCR/SSR, SRS	Gemini Gemini	GMS GMS (no Agena TLM)	Orbit Reentry	Gemini Gemini	GSSC(7) GSSC(7)	Orbit Reentry	N.A.	GMS	N.A.
FIDO Crew Training Simulation	Rendezvous Launch	MOCR Open Loop	MOCR/SSR (Flt.Dyn. SSR)	Gemini	GSSC (no Agena TLM)	Launch	Gemini Agena	GSSC SOC(8)	Launch Orbit	GSSC	GSSC	N.A.
Agena Simulation	Launch Orbit	Integrated Closed Loop	MOCR/SSR, SRS (Agena & Flt.Dyn. SSR)	Agena Agena	GSSC GSSC	Launch Orbit	Gemini Agena Agena	SOC(8) GSSC GSSC	Orbit Launch Orbit	GSSC	N.A.	GSSC
Gem./Agena Launch/ Abort Simulations	Launch Launch/ Aborts	MOCR Closed Loop GLV/Gem. Model Closed Loop	MOCR/SSR, SRS MOCR/SSR	Gemini	GMS (no Agena TLM)	Launch	Gemini Agena	GMS SOC(8)	Launch Orbit	GSSC	GMS	N.A.

(1) Phases emphasized are prelaunch, launch, maneuvers.
(2) Tapes played at CKN to MCCH operational PCMGS.
(3) Through BDA first revolution LOS.
(4) Pre-recorded tapes played at remote sites;stateside passes input by tape at MCCH to operational PCMGS
(5) Starts at CYI first AOS.
(6) TTY messages from MSFN.
(7) From GMS ephemeris data input.
(8) By vector extrapolation loaded into SOC.

LIMITATIONS OF SCATS

In the review of the documentation for past simulations certain limitations of the SCATS became apparent. These limitations were the inability of the SCATS to simulate certain mission events or data because of equipment or program restrictions. These limitations, however, appear to be acceptable to the operators and do not detract from the simulated exercises in any significant way. Some of the more obvious limitations were: a) the deletion of certain remote sites in the SRS sequencing because of the existence of only two SRS's, b) the lack of DCS history at the SRS, c) the absence of dump telemetry at the SRS and, d) the lack of certain site radar information at the SRS. In some cases, special techniques have been devised to overcome these limitations such as the playing back of recorded "real" telemetry to simulate dump telemetry to enable the flight controller to exercise dump telemetry processing procedures. Another example is the use of IBM 7094 computers (GSSC) to generate Apollo telemetry streams in lieu of using one IBM 360/75 which is not yet installed. This problem is discussed in Section II. The use of these techniques, unconventional though they may be, indicate a flexible system which does produce the desired effect. In fact, this employment of improvised techniques and the design of simulation exercises around the limitations of the existing equipments has yielded satisfactory results without any large increases in equipment cost.

BACKGROUND TO SCOPE OF STUDY

The original intent of the phase of the MITRE study was to measure the capacity and effectiveness of the SCATS equipments to perform their assigned tasks. When it became evident that the overall system was generating simulations which were of sufficient quality to deliver the desired data to the operating positions and that measures of the equipment capacities, if taken individually, would mostly parallel those measures found in the analyses of the other systems and would not give an overall view of the SCATS capability, the following approach was considered.

The important questions for investigation in the SCATS seemed to be in the areas of operational requirements and procedures for the use of the system. It was decided therefore to conduct the initial analysis on the use of the SCATS equipments during a typical series of simulations for GT-8 and 9, with the goal in mind to accumulate data on the efficiency of the equipment usage.

The extensive attempts to locate suitable data for this usage analysis showed that the required was non-existent and this condition was documented in MITRE Corporation memorandum No. HO-28, dated 9 May 1966, to Mr. Satterfield of NASA. This memorandum also proposed that the product for the final report would define salient technical questions about SCATS that remain unanswered, specify the data required for an analysis of these questions and detail plans for collecting and recording this data during future simulations.

SCOPE OF STUDY

In line with the recommendations in the aforementioned memorandum, which were accepted by the technical monitor, the following sections of this report will recommend certain analyses that should be conducted to provide answers to questions about the SCATS. The data required to perform these analyses and methods for gathering this data will be specified. The recommended analyses result from the initial study of the SCATS and are included here for consideration by NASA. As study of ASCATS is completed,it is felt that additional analyses, or modifications of the analyses proposed in this report, will be desirable.

PURPOSE OF STUDY

The capability, capacity and effectiveness of the SCATS may be measured by its ability to simulate a particular spaceflight or portion thereof, control the simulation and introduce certain aberrations, thereby familiarizing remote site flight controllers and/or MOCR/SSR personnel with the planned mission, while at the same time training these personnel to deal with unexpected situations.

To perform these simulations in as realistic a manner as possible, within the allotted time before a spaceflight, is the objective of the SCATS.

If there is concern with the present SCATS capability to satisfy future requirements an analysis of how effective SCATS is in providing simulation support would be useful to NASA.

The analysis should consider three aspects of the system:

1. The performance and operational efficiency of the SCATS,
2. The efficiency of the monitoring and control capability of SCATS,
3. The efficiency of the SCATS to simulate a mission environment.

The following sections of this report will discuss the value of the analyses to be performed in each of these categories and suggest methods for gathering data to support them.

SECTION II

RECOMMENDED ANALYSES

PERFORMANCE AND OPERATIONAL EFFICIENCY OF THE SCATS EQUIPMENTS

As previously noted, the equipments in the SCATS are used in many configurations to accomplish the required simulations. Certain modes of operation require either different equipments or different configurations of the same equipment. SCATS equipment usage is defined here as the use made of the SCATS equipments from the time a series of simulated exercises is begun, for a particular spaceflight, until F-4 days (or final simulation). The analysis does not include the use of the equipments from F-4 days until the start of the next simulation nor does it include use of equipments during non-scheduled times. An analysis of equipment usage during a series of simulated exercises would answer the following questions:

1. How much of the scheduled equipment time is used for actual running time? How much is idle or waiting time?
2. Can maintenance or other functions be performed on the equipment during extended holding periods?
3. Would a change in the procedures used during a test allow for more efficient use of the equipment?
4. Can backup procedures be initiated or expanded to make use of idle equipment during a primary equipment failure?

EFFICIENCY OF THE MONITORING AND CONTROL CAPABILITY OF SCATS

The display and control system of the SCATS is used to monitor and control the simulated exercise and observe its impact on the operators in the MOCR/SSR or SRS. It is also used to introduce faults into the simulation thereby exposing the operators to abnormal situations.

An analysis of this area will provide answers to the following questions:

1. Are the displays redundant in any areas where redundancy is not required? Is this redundancy a penalizing cost?

2. Does each display contain sufficient data to provide the operator with the answer to his particular questions or is he forced to observe more than one display to compile the data needed?

3. What use is made of the displays presently available in the SCA area? (i.e., how often is each display used by each operator?)

4. Do the console modules in the SCA consoles permit successful monitoring and control of critical mission situations?

EFFICIENCY OF THE SCATS SIMULATION

This study would determine the efficiency of certain equipment/program combinations in providing simulated exercises. This is not meant to evaluate the quality of the data presently being provided, for this is for the experts in specific disciplines to determine, but rather to investigate and report on how efficiently the present equipment/program combinations are being used to generate the data. The two analyses below are applicable only to SCATS and it may not be desirable to conduct these specific analyses prior to the advent of ASCATS. Analysis of this nature should be considered for ASCATS and will be recommended when study of the ASCATS equipments and programs is complete.

Comparison of Simulated Data vs. "Real-World" Data

This recommended study involves an item-by-item comparison of data presented to the operators during a live mission with that presented during a SCATS generated exercise. The object here is to determine if the data arriving at the MOCR/SSR and or SRS operator observation points (i.e., consoles, recorders, etc.) during a simulation, is accurately portrayed.

We might then ask the questions - is the data portrayed unnecessarily complex for a simulated environment? - and as a result of this complexity does it overtask any of the equipment/program combinations? - for example, would two samples per second of a certain TLM parameter satisfy a simulated environment instead of an operational rate of ten samples per second.

GSSC Telemetry Generation

The percentage of the capacity of the GSSC being used for generation of the TLM bit streams for Apollo is quite high. Statistics supplied by NASA indicate the generation of the bit stream for all vehicles requires over 125% of the operating time of a 7094 computer (two 7094's are presently being used). This time does not include generation of trajectory data, DCS requirements, displays, control or the time employed by the executive program. The TLM streams being generated are in the operational format at the sample rate required for live operation. Unanswered questions in this area are:

1. Would a decrease in the TLM sample rate be practical during simulated exercises thereby permitting more available time in the GSSC?

2. Can more efficient use be made of the GSSC executive program to allow more available processing time in the GSSC?

SECTION III

DATA REQUIREMENTS AND DATA ACQUISITION FOR ANALYSES

DATA REQUIRED FOR EQUIPMENT USAGE ANALYSIS

To answer the questions about the efficiency of the use of the SCATS equipment the following data should be made available for analysis:

a. Logs - a detailed log of the use of the following equipments should be maintained during each simulated exercise:

    SRS Consoles
    Simulated Digital Command System Units (SDCSU)
    SCATS PCM Ground Stations (PCMGS)
    Process Control Unit (PCU)
    Ground Support Simulation Computer (GSSC)
    SCATS Control (SCA) Consoles

b. Log Contents - each log should indicate the following data for each equipment specified:

    status of equipment at start of test,
    time at start of test,
    time of test interruption,
    reason for interruption (delay, hold, failure, program loop, etc.),
    time of each restart,
    time at end of test.

c. Method - all data required should be manually recorded at a central location, such as the simulation supervisor's position, where it is possible to investigate by telephone the reasons for the delays.

The analysis of this data, when compiled for a complete series of simulated exercises, will point out where major delays are encountered and the reasons for them. It is then a simple step to determine if the delays are sufficiently significant to warrant changes in maintenance or operating procedures.

DATA REQUIRED FOR ANALYSIS OF MONITOR AND CONTROL CAPABILITY

Data required for this analysis would be obtained from two sources. The first is computer recorded printouts of all console actions addressing the GSSC in any manner and these printouts should be taken for <u>all</u> modules in the SCATS Control Area (SCA). These recordings should be made for a complete series of simulated exercises to permit an analysis of display and module usage during all phases of a simulated spaceflight.

The second set of data required is a description of all GSSC generated displays available to the SCA consoles and the contents of each display. It should be noted that the documentation made available does not include all displays.

The contents of each GSSC generated display will be studied and a comparison of each display's contents will determine where redundant information exists and if the redundancy is necessary for display continuity.

Next, a study of the display contents will determine if the data presented is adequate for the SCA operator to perform his duties of if he is forced to gather information from other sources because of the lack of information on the display. Finally, an analysis of the printouts of operator actions will determine which displays and modules are commonly used by the SCA operators during the various simulated mission phases (i.e., launch/abort, network simulations, etc.).

Analysis of this composite data could result in recommendations for display changes and/or module changes in the SCA.

SOME DATA REQUIREMENTS TO DETERMINE EFFICIENCY OF SCATS SIMULATIONS

The data required for an analysis of the efficiency of the use of the SCATS equipment/program in providing simulated exercises will of course be dependent on the system to be analyzed (i.e., SCATS or ASCATS).

The analyst must have a general knowledge of the data required by the operators being "trained." One way to provide this knowledge would be to interview certain key MOCR and/or remote site flight controllers to determine if the data being supplied to them by SCATS is satisfactory for simulation purposes. Whereas in an operational environment certain information may be available to an operator in detailed form it may not be necessary to present this in as detailed a manner in a simulation. For example, a certain TLM parameter sampled at 50 times per second in a real environment may be presented at a much lower rate during a simulated mission and still serve the same purpose. It may also be determined, by interviewing the flight controllers, that certain desirable data is not available from SCATS. An example here would be the inability of SCATS to produce certain faults in a spacecraft system that would exercise a particular console operator. Once the minimum acceptable requirements are derived, it is then possible to determine if the data is available from SCATS. If it is not, it must be determined whether the limitation is a result of equipment or program restrictions or both. Exactly how this would be determined would vary for each individual case. The following simple illustration will indicate the result of an investigation of this type.

Certain remote sites are not assigned by the sequencing program of the GSSC to an SRS during a simulation if the time between AOS and LOS is less than a specified minimum period. At present, this is an acceptable limitation of the SCATS. If it should become unacceptable, one possible solution would be to add a third SRS to SCATS, thereby permitting three contingents of flight controllers to operate at the same time.

The data required to perform the analysis of the time required for the GSSC to generate TLM bit streams would be actual statistics on the loading of the GSSC in a maximum simulation case. With these figures we can then determine what impact a lower sample rate, which would decrease the processing time, would have on the rest of the system.

SECTION IV

SUMMARY

The report on SCATS has presented the recommendations for three major areas of investigation, namely, the performance and operational efficiency, the control and monitoring capability and the efficiency of the use of the SCATS simulation capabilities. Certain salient questions have been defined in each area of investigation and the data and methods for compiling the data to answer these questions have been set forth.

It is our opinion that the SCATS is an efficient and flexible system presently capable of simulating most of the data presented to MCCH and remote site flight controllers during all phases of a Gemini/Agena mission to an acceptable degree of accuracy. It has, however, certain limitations which do not appear at present to detract in any significant way from the quality of the simulated exercises.

Future performance analyses should emphasize the areas of operational requirements and procedures. It is from these studies that the most beneficial information would be derived to guide improvements or modifications to the system, rather than detailed performance analyses of individual SCATS equipments.

APPENDIX A

This Appendix contains a bibliography of documents consulted in accomplishing the analysis reported in this Part II of Volume III.

1. Simulation Data Acquisition Requirements Gemini VIII, 30 Dec. 1965.
2. Apollo SCATS System Specification, 8 Dec. 1965.
3. Simulation Control Subsystem, PHO-SM 304, Vols. I and IA.
4. Simulated Remote Sites Subsystem, PHO-SM 302.
5. Simulation Interface Subsystem, PHO-SM 305.
6. Simulation Data Subsystem PHO-SM 303.
7. SCATS Operations document, "Configuration and Initialization PHO-TR145", Vols. I and IA.
8. SCATS Utilization Plan, MOCR Open-Loop Operations, PHO-TR145, Vol. II.
9. MCCH SCATS Operations document PHO-TR145, Vol. III.
10. GTA-8 Remote Site Data Processing Requirements, 3 Feb. 1966.
11. Simulation Control Subsystem Specification, Spec. No. 153300-00069D, 5 June 1965.
12. Simulation Interface Subsystem Specification, Spec. No. 153400-00092C, 20 Jan. 1964.
13. Simulated Remote Sites Subsystem Specification, Spec. No. 153100-00068A, 16 Dec. 1963.
14. Simulation Data Subsystem Specification, Spec. No. 153200-00070D, 24 Feb. 1966.
15. Simulation Real-Time Computer Program Requirements, Single Vehicle, PHO-TR120, Vol. 6A.
16. Simulation Real-Time Computer Program Requirements, Gemini Rendezvous, PHO-TR120, Vol. 6B.
17. Simulation Real-Time Computer Program Requirements, Gemini Launch Vehicle, Gemini PHO-TR120, Vol. 6C.
18. PCU Programming Requirement (Milestone II) Specification 1S3204-01207B, 10 July 1964.
19. FCDAR-MSFN Gemini IX, 4 March 1966.
20. Notes on Gemini VI, Carl Shelley.
21. Notes on Gemini VIII, Carl Shelley.

APPENDIX B

DISTRIBUTION LIST

D-01

R. R. Everett
C. W. Farr
J. F. Jacobs

D-05

R. R. Bennie
D. I. Buckley
A. Goldstein
C. C. Grandy
S. J. Hauser, Jr.
G. A. Hawthorne
E. S. Herndon
K. E. McVicar
J. S. Quilty
W. B. Woodward (20)

D-06

D. R. Israel

D-51

A. Cohen
D. Goldenberg (5)
S. E. Rose

D-84

L. C. Driscoll
F. B. Kapper
M. W. Lodato
J. D. Porter

NASA

K. J. Allen, FS2
D. A. Beckman, FC
C. A. Beers, FC
D. F. Bruce, BG6
H. E. Clements, FS
D. E. Fielder, FA4 (3)
L. R. Hamilton, FS3
J. D. Hodge, FC
C. C. Kraft, FA
E. F. Kranz, FC
H. E. Leech, FS3
J. R. McCown, FS2
J. A. Miller, FS2
D. T. Myles, FS2
D. H. Owen, FC
R. G. Palmer, FS2
L. L. Ruetz, FS3
J. M. Satterfield, FS2 (80)
S. A. Sjoberg, FA
J. G. Zarcaro, FA4